DISTRIBUTED COMPUTING BY MOBILE ROBOTS: GATHERING*

MARK CIELIEBAK†, PAOLA FLOCCHINI‡, GIUSEPPE PRENCIPE§, AND
NICOLA SANTORO¶

Abstract. Consider a set of $n > 2$ identical mobile computational entities in the plane, called robots, operating in Look-Compute-Move cycles, without any means of direct communication. The GATHERING PROBLEM is the primitive task of all entities gathering in finite time at a point not fixed in advance, without any external control. The problem has been extensively studied in the literature under a variety of strong assumptions (e.g., synchronicity of the cycles, instantaneous movements, complete memory of the past, common coordinate system, etc.). In this paper we consider the setting without those assumptions, that is, when the entities are oblivious (i.e., they do not remember results and observations from previous cycles), disoriented (i.e., have no common coordinate system), and fully asynchronous (i.e., no assumptions exist on timing of cycles and activities within a cycle). The existing algorithmic contributions for such robots are limited to solutions for $n \leq 4$ or for restricted sets of initial configurations of the robots; the question of whether such weak robots could deterministically gather has remained open. In this paper, we prove that indeed the GATHERING PROBLEM is solvable, for any $n > 2$ and any initial configuration, even under such restrictive conditions.

Key words. autonomous mobile robots, gathering, asynchrony, unlimited visibility, obliviousness, distributed computing

AMS subject classification. 68W15

DOI. 10.1137/100796534

1. Introduction.

1.1. Setting. In distributed computing, the research focus is on the computational and complexity issues in systems composed of autonomous computational entities interacting with each other (e.g., to solve a problem or to perform a task). While traditionally the entities have been assumed to be *static*, recent advances in a variety of fields, ranging from robotics to networking, have motivated the distributed computing community to address the situation of *mobile* entities located in a spatial universe $\mathcal{U}$. The entities, called *agents* or *robots*, have storage and processing capabilities, exhibit the same behavior (i.e., execute the same protocol), and can move in $\mathcal{U}$ (their movement is constrained by the nature of $\mathcal{U}$). Depending on the nature of $\mathcal{U}$, there are two basic settings in which autonomous mobile entities are being investigated. The first setting, sometimes called *graph world* or *discrete universe*, is when the universe is a simple graph; the second setting, sometimes called *continuous universe*, is when $\mathcal{U}$ is a region of the two-dimensional (2D) space. In this paper we are interested in the continuous setting of mobile computational entities in the plane, which has been investigated by researchers in distributed computing as well as

†sd&m Schweiz AG, 8050 Zurich, Switzerland (mark@dreamboxx.com).

‡School of Electrical Engineering and Computer Science, Ottawa University, Ottawa, ON, K1N 6N5, Canada (flocchin@site.uottawa.ca).

§Department of Informatics, University of Pisa, 56100 Pisa, Italy (prencipe@di.unipi.it).

¶School of Computer Science, Carleton University, Ottawa, ON, K1S 5B6, Canada (santoro@scs.carleton.ca).

in robotics, control, and artificial intelligence, albeit with different assumptions (see, e.g., [1, 3, 11, 13, 21, 26, 27, 28, 29, 30, 33, 41, 40, 43, 49, 51, 52]; for recent surveys see [5, 24, 25, 47]).

In distributed computing, each mobile entity, traditionally called robot, is modelled as a computational unit provided with its own local memory and capable of performing local computations. The robots are placed in the plane, and are viewed as points in $\mathbb{R}^2$. Each robot has its own local coordinate system; however, the local coordinate systems of the robots might not be consistent with each other. A robot is endowed with sensorial capabilities and it observes the world by activating its sensors, which return a snapshot of the positions of all other robots with respect to its local coordinate system. Each robot is endowed with motorial capabilities, and can move freely in the plane. A move may end before the robot reaches its destination, e.g., because of limits to its motion energy. The robots are indistinguishable by their appearance, execute the same protocol, and have no explicit communication capabilities. The robots operate autonomously, without a central control, in *Look-Compute-Move* cycles. During a cycle, a robot obtains a snapshot of the environment (*Look*); executes the protocol, the same for all robots, using the snapshot as an input (*Compute*); and moves toward the computed destination, if any (*Move*). After each cycle, a robot may be inactive for some time. For more details on the model, see the distributed computing literature on the subject (e.g., [1, 3, 11, 15, 24, 26, 27, 46, 49, 51, 52, 56]), as well as section 2 in this paper.

Different (sub)models arise depending on the *additional* assumptions made on the capabilities of the robots and on the nature of the system.

The first basic distinction is about time and level of *synchronization*. In particular, three models are commonly used: FSYNC, SSYNC, and ASYNC; the fully synchronous model FSYNC is the strongest, the asynchronous model ASYNC is the weakest, and the semisynchronous model SSYNC lies in between. In the *synchronous* models (FSYNC and SSYNC), the cycles of all robots are fully synchronized: the sensors that become active do so all at the same time and each operation of the life cycle is performed by all robots simultaneously [1, 6, 11, 13, 19, 20, 21, 29, 41, 52]. In the *asynchronous* model (ASYNC), there is no global clock, and the robots do not have a common notion of time; furthermore, the duration of each activity (or inactivity) is finite but unpredictable [15, 26, 27, 36, 46].

With respect to the level of global agreement on the local *coordinate systems*, different assumptions are made, ranging from availability of a global positioning system [31, 32, 35, 40, 43], to the agreement on the direction and orientation of both axes but not on the unit of distance nor the origin (e.g., as provided by a compass) [26, 27], to partially accurate agreement on the direction and orientation (e.g., inaccurate compass) [34, 36, 49], to the absence of any relationship among the local coordinate systems of different robots [6, 27, 52], i.e., when the robots are *disoriented*.

Two submodels have been identified with respect to workspace *memory* the robots have available and its persistence. In the *oblivious* model, all the information contained in the workspace is cleared at the end of each cycle. In other words, at the beginning of each cycle, the robots have *no* memory of past actions and computations, and the computation is based solely on what is determined in the current cycle. The importance of obliviousness comes from its link to self-stabilization and fault-tolerance. This model, sometimes called memoryless, is used, e.g., in [11, 13, 20, 26, 27]. In the *persistent memory* model, all the information contained in the workspace is legacy: unless explicitly erased by the robot, it will persist thoughout

the robot's cycles [8, 52].

In this paper, we consider the weak setting when the entities are oblivious (i.e., they do not remember results and observations from previous cycles), disoriented (i.e., they have no common coordinate system), and fully asynchronous (i.e., no assumptions exist on synchronization of cycles and timing of activities within a cycle).

1.2. The problem and its difficulty. The problem we consider is the GATHERING PROBLEM: given n robots arbitrarily placed in the plane, with no two robots at the same position, have them gather at the same point in a finite number of cycles; the gathering point is not fixed in advance. This task constitutes one of the very basic primitives for the control and coordination of autonomous mobile robots. Known also as *rendezvous*, *point formation* (because the robots are viewed as points in the plane), *aggregation*, and *homing*, this task has been the object of intensive investigations in robotics, artificial intelligence, control, and distributed computing (e.g., see [1, 3, 8, 11, 16, 55, 26, 29, 30, 36, 41, 43, 44, 45, 46, 48, 52, 56]).

In distributed computing, the GATHERING PROBLEM has been solved under a variety of strong assumptions. In particular, solutions exist assuming (full or partial) *synchronicity of the cycles* and *instantaneous movements* (i.e., in FSYNC and SSYNC), even if the robots are oblivious and disoriented [52], assuming the robots have *unbounded persistent memory* (i.e., they can record all past observations and computations), even if they are asynchronous and disoriented [8], and assuming agreement on the local coordinate systems (i.e., the robots are not disoriented), even if the robots are oblivious, asynchronous, and with limited visibility [26].

In this paper, we ask whether it is possible to solve the GATHERING PROBLEM without these additional strong assumptions; that is, we consider the problem in the weak setting when the entities are oblivious, disoriented, and fully asynchronous. This question has been open for a long time. It is known that multiplicity detection (i.e., the ability to detect whether at a point there is none, one, or more than one robot) is necessary in the absence of additional assumptions [48]; thus, in the following we will assume it.

Notice that the simpler CONVERGE problem, where the robots are required only to move "very close" to each other, without necessarily gathering at the same point [3, 11, 15, 19, 46], can be easily solved: each robot computes the center of gravity of all robots and moves toward it; it has been shown that using this strategy the robots converge toward the same point, albeit without ever reaching it [11]. The reason the same solution (i.e., moving toward the center of gravity) does not work for the GATHERING PROBLEM is because the center of gravity is not invariant with respect to robots' movements toward it and the robots are oblivious. In fact, once a robot makes a move toward the center of gravity, the position of the center of gravity changes; since the robots act independently and asynchronously from each other and have no memory of the past, a robot (even the same one) observing the new configuration will compute and move toward a different point.

The obvious solution strategy for the GATHERING PROBLEM would be to choose as destination a point that, unlike the center of gravity, is *invariant* with respect to the robots' movements toward it. The only known point with such a property is the unique point in the plane that minimizes the sum of the distances between itself and all positions of the robots. In fact, this point, known as the *Weber* (or *Fermat* or *Torricelli*) *point*, does not change when moving any of the robots straight toward it [39, 53]. Unfortunately, it has been proven in [4] that the Weber point is not expressible as an algebraic expression involving radicals since its computation

requires finding zeros of high-order polynomials even for the case $n = 5$ (see also [10]). In other words, for $n \geq 5$, the Weber point is *not computable* even by radicals; thus it cannot be used to solve the GATHERING PROBLEM. Interestingly, even *convergence* toward the Weber point cannot be guaranteed due to its instability with respect to changes in the point set [23].

The existing algorithmic contributions for robots in our setting are limited to solutions for $n = 3$ and $n = 4$ robots or for restricted sets of initial configurations of the robots [9] (the problem is unsolvable for $n = 2$ robots [52]). The lack of results so far on the GATHERING PROBLEM is not due to lack of research efforts, but rather to the inherent difficulties that this weak setting presents. In fact, the simultaneous presence of asynchrony, obliviousness, and disorientation impose severe limitations to the robots to cooperatively perform the assigned task. In particular, their inability to remember the past and the asynchrony of their behavior are crucially hindering factors.

Since robots are oblivious, they do not have any memory of past observations, and the destination is decided by a robot during a *Compute* operation solely on the basis of the location of other robots perceived in the last *Look* operation. Asynchrony implies that, based on an observation made at some time t, a robot r computes a destination at some time $t' > t$, starts to move to its destination at an even later time $t'' > t'$, eventually stopping at time $t''' \geq t''$; thus it might be possible that at time t'' some robots are in different positions from those previously perceived by r at time t, because in the meantime they performed their *Move* operations (possibly several times). In other words, robots may move based on significantly outdated perceptions.

Among the many difficulties created by this fact is the difficulty of avoiding *collisions*: since the robots do not look while moving (the robot's sensors are activated only during the *Look*), and the destination is computed based on possibly outdated information about the position (and moves) of the other robots, to avoid collisions, the computation of a robot r must take into account all possible movements of all the other robots from the time t of the *Look* to the *unknown* and a priori *unbounded* time $t''' > t$ when r will actually end its move. In other words, collision avoidance, if required, is difficult, and it is the sole responsibility of the protocol.

An additional difficulty due to obliviousness and related to collisions is that if two robots (accidentally or by design) end a cycle at the same location, then they become indistinguishable, and from that moment on they might behave exactly in the same way (in fact, there is at least one execution in which they will do so); in particular, it might not be possible for them to separate ever again.

In this paper, we prove that, in spite of all these difficulties, the GATHERING PROBLEM is solvable for any $n > 2$ and any initial configuration, even under the restrictive conditions of asynchrony, obliviousness, and disorientation.

1.3. Our solution. The overall strategy followed by the robots is quite simple to state: at the beginning the robots are in distinct locations (forming a *plain* configuration); within finite time, a unique *dense* point (i.e., where there is more than one robot) is created, and all other robots gather there. However, since the robots are disoriented and oblivious and operate in a totally asynchronous manner, this strategy is not simple to enact. For example, ensuring that a unique dense point is created requires that during the execution of the algorithm no collisions occur at any point other than the final gathering one. An additional difficulty is in recognizing if a symmetric configuration is being formed during the execution. For instance, if all the robots initially are the vertices of an n-gon (a configuration called *equiangular*), then the trivial

strategy in this case would be that the robots move toward the center of the n-gon; however, if such a configuration is created by the movement of some robots during the execution, the still robots might observe the equiangular configuration and decide to apply the go-to-center strategy, while those already moving continue their procedure (possibly destroying the newly formed equiangularity). Our algorithm ensures that, if a symmetric configuration is formed during the execution, *all* robots become aware of it (recall, however, that the robots are oblivious and do not remember previous observations), so that *all* robots follow the same strategy.

The algorithm works by examining the configuration observed by a robot in the *Look* operation. The first test a robot does when computing is to determine whether there is a single dense point, p; if so, the robot moves toward p. In absence of a dense point, the robot checks for the presence of a specific symmetric configuration: the *biangular* configuration. If the check for a biangular configuration is positive, the robot will move toward the center of biangularity, b. The algorithm ensures that, if this case is recognized by one robot, then *all* robots will recognize it and will move toward b; in this case, within finite time b will become dense.

Should the first two tests fail, the robot analyzes the *string of angles* of the robots with respect to the center c of the smallest enclosing circle. The algorithm distinguishes four cases. For all of them, the algorithm uses the string of angles of the robots to "elect" a subset of the robots. If the elected set consists of a single robot, that robot moves until it reaches another robot, thus creating a single dense point.

Otherwise, the robots of the elected set move toward c, ensuring that the smallest enclosing circle is not changed by their movements, and paying particular attention to *potential* biangular configuration that might be formed during their movements. In fact, it is possible that the elected robots reach, during their movements, points that render the configuration biangular; such points are called *critical*. The algorithm explicitly computes these points; in particular, if an elected robot has a critical point on its way, the algorithms ensure that it reaches it; also, the algorithm ensures that the configuration is *still* (i.e., no robot is moving or about to move) when this happens. Hence, if a biangular configuration is formed during the movements of the elected robots, all other robots will observe it in their next *Look* state and will eventually gather on the center of biangularity, as described above. Otherwise, if no biangular configuration is formed, the elected robots will create a unique dense point at c, where all other robots will gather.

1.4. Related work. Gathering a set of autonomous mobile robots dispersed in the plane constitutes a basic control and coordination task. In distributed computing, this problem has been extensively studied in a variety of settings.

The most difficult setting for the gathering problem for oblivious robots is clearly the *asynchronous* one (ASYNC), where no timing assumptions are made; as mentioned earlier, the only solutions are for $2 < n < 5$, and for restricted sets of initial configurations of the robots when $n \geq 5$ [9]. A solution protocol has been presented also in the case of limited visibility, provided there is agreement on the coordinate system (e.g., a compass) [26]. Probabilistic protocols for gathering in absence of agreement on the coordinate systems have been proposed and experimentally analyzed in [50].

In the *semisynchronous* setting SSYNC, the gathering problem of oblivious robots has been tackled in [11, 41, 44]. With limited visibility, a solution has been proposed in [3], where the robots *converge* toward the same point.

There have been several investigations on the GATHERING PROBLEM with robots operating in the *fully synchronous* setting FSYNC [29, 41, 44, 57]. The starting point

of these investigations is the convergence protocol of [3], operating in the SSYNC and thus in the FSYNC models. Like [3], these protocols work for oblivious robots with no common coordinate system and limited visibility, and they all converge toward a unique point; unlike [3], they are only for the FSYNC model.

The *nonoblivious* case has been studied in [8], where a protocol to achieve gathering in ASYNC with unbounded memory in finite time has been presented. Convergence of nonoblivious robots toward a single point has also been achieved with limited visibility under a restricted form of asynchrony [41].

The investigations have also considered the case of "fat" robots, that is, when the robots are not considered to be points but rather discs. The study of gathering in this case has considered only a few robots [7, 14, 16].

The problem has also been examined when there are robot failures or movement inaccuracies in stronger models or on restricted spaces [1, 12, 55, 30]. Also investigated has been the gathering with a compass that might be inaccurate (e.g., [36, 49]); this setting is stronger than the one considered here, which does not assume the availability of any compass.

The gathering problem is part of the more general class of problems called *Pattern Formation* which requires the robots to move in the plane to form a prescribed pattern given in input. Specific important patterns are precisely the point (i.e., gathering), the line [13], and the circle (e.g., [18, 22]). Special attention has been given to *Arbitrary Pattern Formation*, the problem of forming *any* pattern given in input (e.g., see [17, 27, 52, 56]).

It is interesting to note that gathering by asynchronous, oblivious, disoriented, and anonymous robots has been studied also when the universe $\mathcal{U}$ is discrete, i.e., when it is a graph (e.g., [37, 38]).

1.5. Organization. The paper is organized as follows. In section 2 the definitions and terminology are introduced, basic geometric properties are established, and techniques for detecting some geometric features are described; the proofs of some of these properties can be found in Appendix A. In section 3, the notion of the critical point, crucial for our solution, is introduced, and, based on this notion, the moving primitives that will be used in our algorithm are defined in section 4. The algorithm that solves the GATHERING PROBLEM for any arbitrary initial configuration is described and its correctness analyzed in section 5. Finally, some open research problems are outlined in section 6.

2. Model and basic properties. In this section, we introduce the model of robots in use and define the basic concepts that we need to present our algorithm for the GATHERING PROBLEM.

2.1. Autonomous mobile robots. The system is composed of a set $\mathcal{R} = \{r_1, \ldots, r_n\}$ of $n \geq 5$ mobile robots,[1] each modeled as a computational unit provided with its own local memory and capable of performing local computations.

A robot is endowed with sensorial capabilities, and it can perceive the spatial environment and the robots in it. Each robot has its own local coordinate system: a unit of length, an origin, and a Cartesian coordinate system defined by the directions of two coordinate axes, identified as the x and y axes, together with their orientations, identified as the positive and negative sides of the axes. However, the local coordinate systems of the robots might not be consistent with each other.

---

[1]For the cases $2 < n < 5$, solution protocols already exist [9].

The robots are *anonymous*: they are indistinguishable by their appearance and without identifiers that can be used during the computation. The robots are *autonomous*, without a central control.

Each robot is endowed with motorial capabilities; it can turn and move in any direction. A move may end before the robot reaches its destination, e.g., because of limits to its motion energy. The distance traveled in a move is neither infinite nor infinitesimally small. More precisely, there exists an (arbitrarily small) constant $\delta_r > 0$ such that if the destination point is closer than δ_r, r will reach it; otherwise, r will move toward it of at least δ_r. We shall refer to this restriction as *Assumption* DIS. Note that, without this assumption, an adversary would make it impossible for any robot to ever reach its destination, following a classical Zenonian argument. In the following, we shall use $\delta = \min_r \delta_r$.

The robots are *silent*: they have no means of direct communication of information to other robots. Thus, any communication occurs in a totally implicit manner, by observing the other robots' positions. Each robot is viewed as a point: let $r(t)$ denote the position of robot r at time t; when no ambiguity arises, we shall omit the temporal indication.

The robots execute the same deterministic algorithm, which takes as input the observed positions of the robots and returns a destination point toward which the executing robot moves.

At any point in time, a robot is either *active* or *inactive*. When *active*, a robot r performs the following three operations, each in a different state:

(i) *Look*. The robot observes the world by activating its sensors, which return a snapshot of the positions of all other robots with respect to its local coordinate system (since robots are viewed as points, their positions in the plane are just the set of their coordinates). A robot can detect whether at a point there is none, one, or more than one robot; i.e., it can detect multiplicities.

(ii) *Compute*. The robot performs a *local computation* according to its algorithm. The result of the computation is a destination point; if this point is the current location, the robot stays still (performs a *null movement*).

(iii) *Move*. The robot moves toward the computed destination; this operation can terminate before the robot has reached it.

When *inactive*, a robot is in a *Wait* state.

(iv) *Wait*. The robot is idle. A robot cannot stay infinitely idle.

A robot is initially in a *waiting* state (*Wait*); the sequence: *Wait-Look-Compute-Move* is called a *computation cycle* (or briefly *cycle*) of a robot. Concerning time, we will assume that the amount of time required by a robot r to complete a computational cycle is finite. As no other assumption on time exists, the resulting system is fully *asynchronous* and the duration of each activity (or inactivity) is unpredictable. This is precisely the ASYNC model. As a result, the robots do not have a common notion of time, robots can be seen while moving, and computations can be made based on obsolete observations.

A remark is needed regarding the *Look* state. As already stated, the result of this state is a set of positions retrieved at one time instant, i.e., at the time when the snapshot of the world was done. Thus, each *Look* can be split into three parts: in the first part the sensors are activated; in the second part the actual snapshot is performed; and in the last part, the data captured by the sensors are sent away in order to be processed. In the following, we shall assume that the first and third parts have null length. This is not a loss of generality: in fact, the first part can be thought
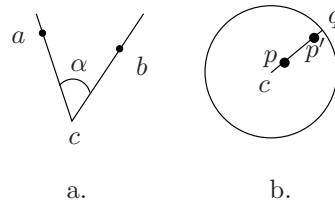
a.   b.

FIG. 2.1. (a) *Convex angle $\alpha = \sphericalangle(a, c, b)$. (b) Two points, p and p', on the same radius.*

to be part of the previous *Wait* state and the third part of the following *Compute* state; therefore, each *Look* coincides with the snapshot. According to this assumption, if r is executing a *Look* at time t, then its view of the world is the snapshot retrieved at t.

Notice that the only time the sensors are activated (i.e., the robot observes the environment) is during *Look*; in particular, a robot does not sense the environment while moving.

The robots that, at time t, are moving or are computing a nonnull movement are said to be *acting* at time t. Furthermore, we say that the acting robots at time t are *acting on p* if they are moving toward point p or their computed destination is p.

We shall partition the robots into sets depending on their state at a given time. Let $\mathbb{W}(t)$, $\mathbb{L}(t)$, $\mathbb{C}(t)$, and $\mathbb{M}(t)$ denote the sets of all the robots that at time t are, respectively, in states *Wait*, *Look*, *Compute*, and *Move*. Let the subset $\mathbb{C}_\emptyset(t) \subseteq \mathbb{C}(t)$ contain those robots whose computation's result is to execute a *null movement*, and let the subset $\mathbb{M}_\emptyset(t) \subseteq \mathbb{M}(t)$ contain the robots executing a *null movement*. We say that a robot r is *still* if $r \in \mathbb{L}(t) \cup \mathbb{C}_\emptyset(t) \cup \mathbb{M}_\emptyset(t) \cup \mathbb{W}(t)$.

A *configuration* (of the robots) at time t, denoted by $\mathbb{D}_t$, is the set of robots' positions at time t. We say that a configuration $\mathbb{D}_t$ is *still* at time t if all the robots are still at time t; given a subset A of the robots, we say that $\mathbb{D}_t$ is `stillBut`(A) at time t if all robots not in A are still at that time.

A point in the plane is called *dense* if it is occupied by more than one robot; we call *plain* a configuration with no dense point, that is, a configuration where all robots occupy at time t distinct positions. A configuration is *final* at time t if it is still and if there exists a point p_g such that $r_i(t) = p_g$ for all $1 \leq i \leq n$; in this case we say that the robots *have gathered* on point p_g at time t.

We study the problem of gathering the robots into a single point whose location is not predetermined, that is, of transforming a plain configuration into a final one. A *gathering algorithm* is a deterministic algorithm that brings the robots in the system to a final configuration in a finite number of cycles from any given plain configuration.

2.2. Geometric definitions and properties. In this section we introduce the notation and the basic geometric properties that will be used in the rest of the paper. All the proofs of this section can be found in Appendix A.

2.2.1. Basic notation. In the rest of the paper, the following notation is used. Given two distinct points a and b in the plane, $\overrightarrow{ab}$ denotes the half-line that starts in a and passes through b, and $\overline{ab}$ denotes the line segment between a and b. Given two half-lines $\overrightarrow{ca}$ and $\overrightarrow{cb}$, we denote by $\sphericalangle(a, c, b)$ the convex angle (i.e., the angle which is at most 180°) centered in c and with sides $\overrightarrow{ca}$ and $\overrightarrow{cb}$ (Figure 2.1(a)).

Given a circle $\mathcal{C}$ with center c, radius *Rad*, and a point p, we say that *p is on $\mathcal{C}$* if $dist(pc) = Rad$, where $dist(ab)$ denotes the Euclidean distance between point a and
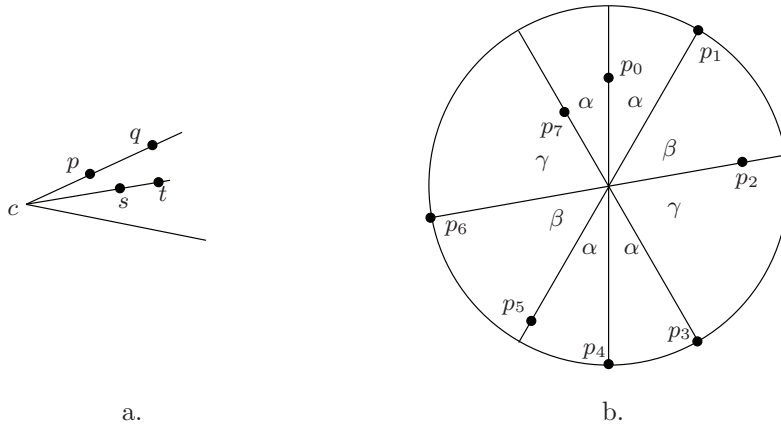
FIG. 2.2. (a) *In the example, $q = \mathtt{succ}(p,c)$ and $s = \mathtt{succ}(q,c)$. (b) Example of the string of angles of $P = p_0, \ldots, p_7$, computed with respect to their SEC, with a clockwise orientation of the circle. We have $SA^+(P,c)[0] = \langle \alpha, \beta, \gamma, \alpha, \alpha, \beta, \gamma, \alpha \rangle$; $LMS(P,c) = \langle \alpha, \alpha, \beta, \gamma, \alpha, \alpha, \beta, \gamma \rangle$; $StS^+(P,c) = \{p_3, p_7\}$; and $StS^-(P,c) = \emptyset$. The points are numbered according to routine $\mathtt{succ}()$.*

b (i.e., p is on the circumference of $\mathcal{C}$); if $dist(pc) < Rad$, we say that p *is inside* $\mathcal{C}$. Given two distinct points p and p', with p inside $\mathcal{C}$, let q be the intersection between the circumference of $\mathcal{C}$ and $\overrightarrow{cp}$. We say that p and p' are on the same radius if $p' \in \overline{cq}$ (see also Figure 2.1(b)). Moreover, we denote by $Rad(p)$ the radius $\overline{cq}$ where p lies; in the following we refer to $Rad(p)$ also as the *radius of p*.

Given points p, p', and p'', the triangle with these three points as vertices is denoted by $\triangle(p, p', p'')$. We use $q \in \triangle(p, p', p'')$ to indicate that q is inside the triangle or on its border.

Given a set of $n \geq 2$ distinct points P in the plane, we denote by $SEC(P)$ (or SEC if set P is unambiguous from the context) the *smallest enclosing circle* of the points; that is, $SEC(P)$ is the circle with minimum radius such that all points from P are inside or on the circle. An example of SEC of a set of eight points is depicted in Figure 2.2(b). The smallest enclosing circle of a set of n points is unique and can be computed in polynomial time [54]. Obviously, the smallest enclosing circle of P remains invariant if we remove all or some of the points from P that are inside $SEC(P)$. Also, we have the following property.

PROPERTY 1. *Given a set P of $n \geq 3$ points, there exists a subset $S \subseteq P$ such that $|S| \leq 3$ and $SEC(S) = SEC(P)$.*

Given a set of points $P = \{p_1, \ldots, p_n\}$ and a point x in the plane, we define the *Weber distance* between x and P by $WD(x, P) = \sum_{p \in P} dist(p, x)$. A point w is the *Weber point* of point set P if it minimizes the Weber distance between P and any point x in the plane, i.e., if $WD(w, P) = \min_{x \in \mathbb{R}^2} WD(x, P)$. If the points in P are not on a line, then the Weber point always exists, and it is unique [53]. Moreover, it is easy to verify (see also [2]) that the following holds.

PROPERTY 2. *Given a set of points P and any point $p \in P$, the Weber point of P is invariant under straight movement of p toward or away from it, with all movements on the half-line connecting the Weber point and p.*

Clearly, the above property holds also for any subset of points in P; this implies that the Weber point might yield a solution for the GATHERING PROBLEM. Unfortunately, it is not computable in general—not even with radicals [4, 10].
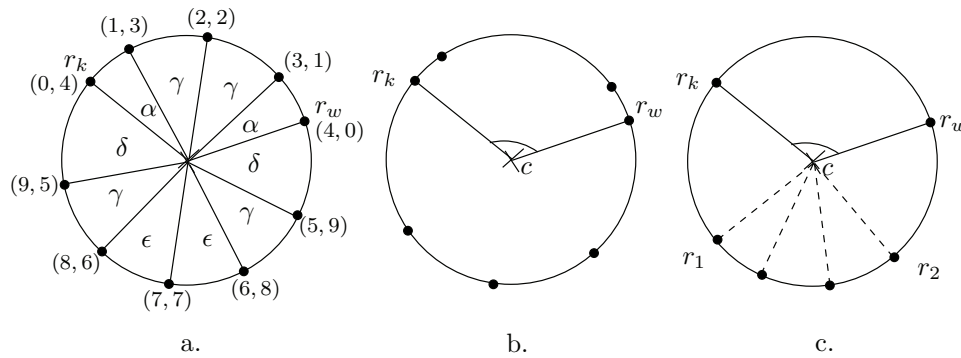
FIG. 2.3. (a) $r_k \in StS^+(\mathcal{R})$, $r_w \in StS^-(\mathcal{R})$, and $LMS(\mathcal{R}) = \langle \alpha, \gamma, \gamma, \alpha, \delta, \gamma, \epsilon, \epsilon, \gamma, \delta \rangle$. Each pair of numbers represents the ranking of the robot in the clockwise (starting from r_k) and in the counterclockwise (starting from r_w) orientation, respectively. (b) and (c) represent two cases described in Lemma 2.2.

2.2.2. String of angles. Given a set P of n distinct points in the plane and a point $c \notin P$ called *center*, let $\bigcup_i \overrightarrow{cp_i}$ be the set of all *rays* starting from c and passing through each $p_i \in P$. The successor of $p \in P$ with respect to c, denoted by $\texttt{succ}(p, c)$, is defined as the point $q \in P$ such that (refer to Figure 2.2(a))

- either q is the closest point to p on the ray where p lies, with $dist(c, q) > dist(c, p)$, if such a point exists;
- or $\overrightarrow{cq}$ is the ray following $\overrightarrow{cp}$ in the order implied by the clockwise direction, and q is the closest point to c on $\overrightarrow{cq}$.

Symmetrically, given a point $q \in P$, the predecessor of q with respect to c, denoted by $\texttt{pred}(q, c)$, is the point $p \in P$ such that $\texttt{succ}(p, c) = q$.

The functions $\texttt{succ}()$ and $\texttt{pred}()$ define a unique cyclic order on P, which we shall denote by $\langle p_0, p_1, \ldots, p_{n-1} \rangle$, where $p_{i+1} = \texttt{succ}(p_i)$; here and in the following, all operations on the indices are modulo n. This in turn defines a cyclic *string of angles* $SA^+(P, c) = \langle \alpha_0, \alpha_1, \ldots, \alpha_{n-1} \rangle$, where $\alpha_i = \triangleleft(p_i, c, p_{i+1})$; p_i is called the (clockwise) *start point* of α_i. The string of angles in the opposite direction is denoted by $SA^-(P, c) = \langle \alpha_{n-1}, \ldots, \alpha_0 \rangle$.

Associated to the cyclic string of angles $SA^+(P, c)$ there is the set of strings $SA^+(P, c)[i] = \langle \alpha_i, \alpha_{i+1}, \ldots, \alpha_{i+n-1} \rangle$, with $0 \leq i \leq n - 1$ (refer to the example depicted in Figure 2.2(b), where the string of angles are computed with respect to the *SEC* of the eight points); similarly, associated to $SA^-(P, c)$ there is the set of strings $SA^-(P, c)[i] = \langle \alpha_{i-1}, \ldots, \alpha_i \rangle$. We define the *start point* of $SA^+(P, c)[i]$ as the start point of α_i, that is, p_i. Finally, let $SA(P, c)[i] = SA^+(P, c)[i] \cup SA^-(P, c)[i]$ and $SA(P, c) = \bigcup_i SA(P, c)[i]$.

We say that $SA(P, c)$ is *simple* if $SA^+(P, c)$ does not contain any angle of zero degrees; otherwise, at least two points are on the same radius, and we say that $SA(P, c)$ is *mixed*.

We denote by $LMS(P, c)$ the *lexicographically minimum string* among all strings in $SA(P, c)$. Let $StS^+(P, c) = \{p_i \in P | SA^+(P, c)[i] = LMS(P, c)\}$ be the set of start points of $LMS(P, c)$ in $SA^+(P, c)$, and let $StS^-(P, c)$ be defined similarly. Let $StS(P, c) = StS^+(P, c) \cup StS^-(P, c)$.

An interesting property of $LMS(P, c)$ is the following lemma.

LEMMA 2.1. *Let $p_k \in StS(P, c)$ be a starting point for $LMS(P, c)$. If $\alpha_i \neq \alpha_j$ for some $0 \leq i, j \leq n - 1$, then $SA^+(P, c)[k] \neq SA^-(P, c)[k]$.*
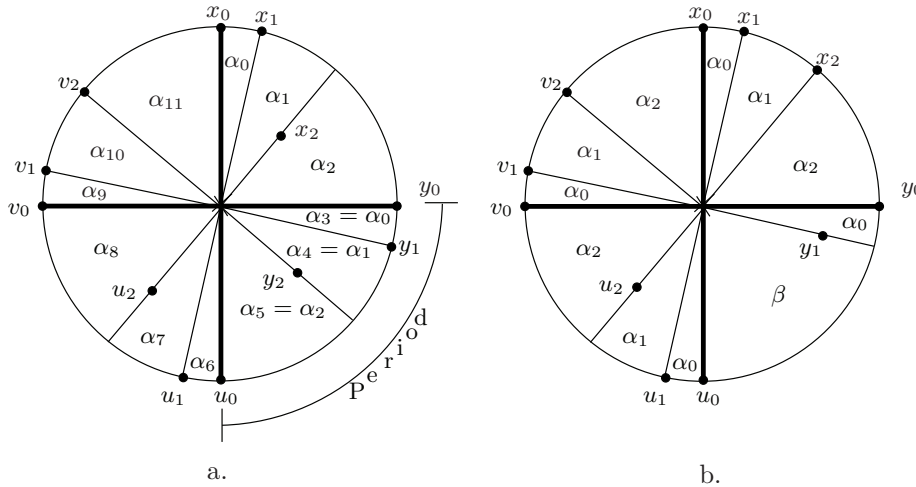
FIG. 2.4. (a) *Example with* $|StS^+(P,c)| = 4$, $LMS(P,c) = \langle \alpha_0, \ldots, \alpha_{11} \rangle$ *with period* $\langle \alpha_0, \alpha_1, \alpha_2 \rangle$. *There are* $\frac{n}{k} = \frac{12}{3} = 4$ *periods, with* $\beta = 90°$. *The thick lines represent the starting points of each of the four periods. Robots* x_i, y_i, u_i, *and* v_i, $0 \le i \le 2$, *are* equivalent. (b) *If* y_2 *is removed from P, we obtain an example of a set of points that is periodic with one gap, with* $\beta = \alpha_1 + \alpha_2$.

In the following, three particular settings related to the string of angles will be of interest: the cases when $s = 1$, $s = 2$, and $s > 2$. In each case, it is possible to establish an ordering of the robots.

When $s = 1$, there is a unique starting position r_k for $LMS(\mathcal{R})$, and by Lemma 2.1, $SA^+(\mathcal{R})[k] \neq SA^-(\mathcal{R})[k]$; without loss of generality, let $LMS(\mathcal{R}) = SA^+(\mathcal{R})[k]$. This yields a total ordering of the robots according to the direction of $LMS(\mathcal{R})$ and starting from r_k; this ordering will be used to achieve different means.

When $s = 2$, $LMS(\mathcal{R})$ does not yield a total ordering of the robots, because there are two start points, r_k and r_w. However, by Lemma 2.1, $SA^+(\mathcal{R})[k] \neq SA^-(\mathcal{R})[k]$ and $SA^+(\mathcal{R})[w] \neq SA^-(\mathcal{R})[w]$; that is, from each start position, $LMS(\mathcal{R})$ is present only in one direction. Based on this property, we can group robots in *teams* of at most two elements and totally order the teams as follows:

1. Rank each robot r according to the total ordering implied by $LMS(\mathcal{R})$ starting from r_k.
2. Rank r according to the total ordering implied by $LMS(\mathcal{R})$ starting from r_w.
3. Assign to r a value v set to the minimum of the two ranks.
4. The robots with the same value form a team.

It is easy to see that this assignment of values has the useful property that each team is composed of either one or two robots (refer to Figure 2.3.a). We will use the total ordering of the teams, according to the assigned values, to achieve different goals. For this case, we have the following interesting property.

LEMMA 2.2. *If all robots are on SEC, there exists a team composed of two distinct robots such that SEC remains invariant if they are removed from SEC.*

Finally, when $s > 2$, the robots can be grouped into *classes*, as described in the next section.

2.2.3. Periodic set of points. We say that a set P of n points is *regular periodic* (or simply *periodic*) if $SA^+(P,c)$ is a periodic string with period greater than or equal
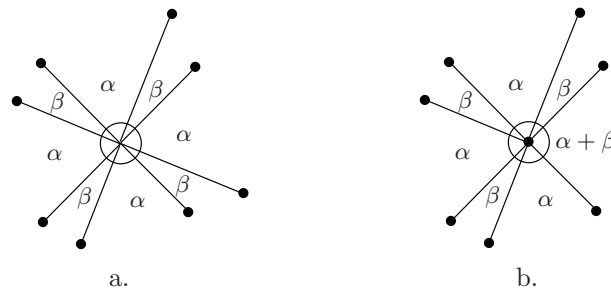
FIG. 2.5. (a) *A regular biangular and* (b) *an irregular biangular set of eight points.*

to 3, where c is the center of the SEC of P; that is, there exist a string W, with $|W| \geq 3$, and $e \geq 2$ such that $SA^+(P,c) = W^e$. Let $s = |StS(P,c)|$, and, without loss of generality, let $|StS^+(P,c)| = k \geq |StS^-(P,c)|$, with $k \geq 2$.

LEMMA 2.3. *Let* $s > 2$. *Then*
  1. $SA^+(P,c)$ *is periodic with period* $\gamma = \frac{n}{k}$, *and*
  2. *either* $|StS^+(P,c)| = |StS^-(P,c)|$ *or* $|StS^-(P,c)| = 0$.

In the following, we also say that $SA(P,c)$ is periodic whenever $SA^+(P,c)$ is periodic. We say that two points $p \in P$ and $p' \in P$ are *equivalent* (modulo periodic shift) if $\sphericalangle(p,c,p')$ is a multiple of $\beta = 360° \cdot \frac{k}{n}$. From Lemma 2.3, the following property holds (refer to Figure 2.4(a)).

PROPERTY 3. *Let* $s > 2$, *and let all points in* P *be on* $SEC(P)$, *and* $p \in P$. *Then,* P *has* $\frac{n}{k} - 1$ *equivalent points. Moreover,* p *and all the points equivalent to* p *form a regular* $\frac{n}{k}$-*gon, and* c *is inside this* $\frac{n}{k}$-*gon.*

All points can thus be partitioned into $\frac{n}{k}$ equivalence classes, all of the same size. Let $EQ(p)$ denote the equivalence class of p. Thus, it is possible to create a total ordering of the equivalence classes of the points according to the period. Two other useful properties will be now introduced.

LEMMA 2.4. *Let* $s > 2$, *and let all points in* P *be on* $SEC(P)$. *Then, for any robot* $p \in P$, $SEC(P)$ *remains invariant if all points except those in* $EQ(p)$ *are removed.*

The other important property is that points can be moved toward SEC or toward c without obtaining set of points that are biangular, as shown by the following lemma.

LEMMA 2.5. *Given a periodic and nonbiangular set of distinct points* P, *then*
  1. *moving any subset of the points toward* $SEC(P)$ *or*
  2. *moving any subset of the points toward* c
does not make the set of points become biangular.

Similarly to the "gaps" introduced for a biangular set of points, we say that a set P of $n - 1$ points is *periodic with one gap* if there exist a string W, with $|W| \geq 3$, and $e \geq 2$ such that $SA^+(P,c) = W^{e-1} \circ W'$, with $W = \langle w_0, \ldots, w_{n/e-1} \rangle$ and $W' = \langle w_0, w_1, \ldots, w_{i-1}, \overline{w}, w_{i+2}, \ldots, w_{n/e-1} \rangle$ for some $0 \leq i \leq n/e - 1$, and with $\overline{w} = w_i + w_{i+1}$ (refer to Figure 2.4(b)). Note that, since $n \geq 5$ and $e \geq 2$, if P is periodic with one gap, then i is unique.

Furthermore, we say that a set P of n points is *irregular periodic* if one of the points in P is at c, and $P \setminus \{c\}$ is periodic with one gap. Note that, the same total order defined for a regular periodic set of points also applies to an irregular periodic set of points.

2.2.4. Biangular set of points. Informally, a biangular configuration is a configuration where there exist a point b and an ordering of the robots such that the

angle between two adjacent robots with respect to b is either α or β, and α and β alternately (refer to Figure 2.5(a)).

More precisely, we say that a set of n distinct points in the plane P is *regular biangular* (or simply *biangular*) if there exists a point b such that for all $i \geq 0$ $\alpha_i = \alpha_{i+2} > 0$ where $SA^+(P, b) = \langle \alpha_0, \ldots, \alpha_{n-1} \rangle$ (refer to Figure 2.5(a)); b is then called the *center of biangularity* of P. Note that, by definition, if P is *regular biangular*, then P has an even number of robots. Note that the equiangular case (i.e., all angles in $SA^+(P, b)$ are the same) is considered here to be just a particular case of the biangular one.

There is a strict relationship between a set of points that is regular biangular and the Weber point of P. In fact, the following property holds [2].

PROPERTY 4. *If P is rotational symmetric with respect to some rotation center z, then z is the Weber point of P. Furthermore, if P is regular biangular, then the Weber point is the center of biangularity.*

Given a set P of $n-1$ points on the plane, we say that P is *regular biangular with one gap* and center b if there exists a point $x_g \notin P$ such that $P \cup \{x_g\}$ is *regular biangular* with center b. In such a case, we call x_g a *gap point* for b; we denote by $\uparrow x_g$ the half-line starting from the center of biangularity of $P \cup \{x_g\}$ and passing through x_g, and we call it a *gap ray*.

Given a set P of $n-2$ points, we say that P is *regular biangular with two gaps* and center b if there exist two distinct points x_g and y_g not in P such that $P \cup \{x_g, y_g\}$ is *regular biangular* with center b. In this case, we call x_g and y_g *gap points* for b; we denote by $\uparrow x_g$ and $\uparrow y_g$ the half-lines starting from the center of biangularity of $P \cup \{x_g, y_g\}$ and passing through x_g and y_g, respectively, and we call them *gap rays*.

Finally, given a set P of n points, we say that P is *irregular biangular* if there exists a point $p \in P$, the center, such that $P \setminus \{p\}$ is *regular biangular with one gap* with center p (refer to Figure 2.5(b)). First we have that, if P is irregular biangular, then its center is unique.

LEMMA 2.6. *Let P be a set of $n \geq 3$ distinct points. If P is irregular biangular, then the center of biangularity is unique.*

In the following, we introduce two simple but useful properties related to biangular sets of points with gaps; let $\gamma = \alpha + \beta$.

PROPERTY 5. *Let P be a set of $n-1$ points on the plane. If P is biangular with one gap and center b, then $\exists S \subset P$, $|S| = n/2$, such that $SA^+(S, b) = \gamma^{\frac{n}{2}}$. That is, S is equiangular.*

PROPERTY 6. *Let P be a set of $n-2$ points. If P is biangular with two gaps and center b, then one of the following two conditions holds:*
1. *There exists a set $S \subset P$ of points, $|S| = n/2 - 1$, such that $SA^+(S, b) = \langle \delta, \gamma^{\frac{n}{2}-2} \rangle$, with $\delta = 2\gamma$. That is, S is equiangular with one gap.*
2. *There exists a set $S \subset P$ of points, $|S| = n/2$, such that $SA^+(S, b) = \langle \gamma^{\frac{n}{2}} \rangle$. That is, S is equiangular.*

Given a set of points P and a cyclic sequence of angles $\sigma = \langle \alpha_0, \ldots, \alpha_{n-1} \rangle$, using the algorithm described in [2], it is possible to efficiently determine whether there exists a point c such that $\sigma \in SA(P, c)$. In other words, there exists an efficient way to determine whether P is biangular if the two angles α and β are given. Routine 7, reported in Appendix C, shows how to determine whether a set of points P is biangular with at most two gaps, or irregular biangular, when the two angles *are not given*; if so, it finds the set B of all centers.
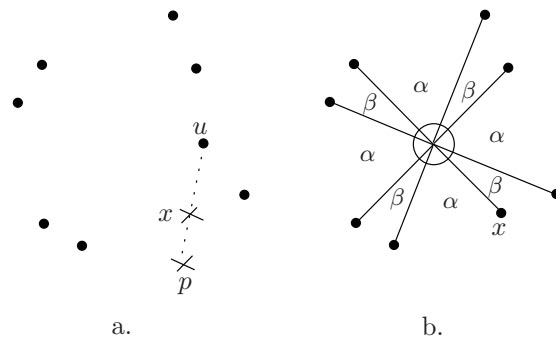
FIG. 3.1. *Critical point:* (a) *x is a critical point for u; in fact* (b) *if u is moved to x, the configuration becomes regular biangular.*

3. Critical points. In this section we introduce the notion of *critical points*, which will be crucial to achieving the final gathering. Informally, a critical point is a point that, if crossed by a moving robot, might at that time generate a biangular configuration (see the example depicted in Figure 3.1).

DEFINITION 3.1. *Given a set of n points P, a point $u \in P$, and another point p (that might or might not belong to P), a point $x \notin P$ is a* critical point *for u with respect to p if either* (i) $x = p$, *or* (ii) $x \in \overline{up}$ *and* $P \setminus \{u\} \cup \{x\}$ *is regular biangular.*

Let us denote with $\mathcal{CP}(u, p)$ the set of critical points of u with respect to p. The first part of Routine 1 (`Routine CriticalOne()`) describes how to compute this set. In particular, Property 5 is exploited as follows: we consider the set $P \setminus \{u\}$ and determine if it is biangular with one gap by invoking Test 2 in Routine 7. If it is not, the configuration cannot become biangular regardless of the position of u (that in this case represents one of the robots). Otherwise, a biangular configuration is created if and only if $\overline{up}$ crosses one of the gap rays. Therefore, the set of critical points of u with respect to p is the set of the points in the intersection between $\overline{up}$ and those rays.

We now extend the previous definition to pairs of points.

DEFINITION 3.2. *Given a set of n points P, two points $u, v \in P$, and another pair of points p', p'' (that might or might not belong to P), a pair of points $x \notin P$ and $y \notin P$ is a pair of* critical points *for u and v with respect to p' and p'' if either* (i) $x = p'$ *and* $y = p''$, *or* (ii) $x \in \overline{up'}$, $y \in \overline{vp''}$, *and* $P \setminus \{u, v\} \cup \{x, y\}$ *is regular biangular.*

Let us denote with $\mathcal{CP}((u, p'), (v, p''))$ the set of critical points of u and v with respect to p' and p''. It follows directly from Definitions 3.1 and 3.2 that the following holds.

PROPERTY 7. $\mathcal{CP}((u, p'), (v, p'')) \subseteq \mathcal{CP}((q', p'), (q'', p''))$, *with* $q' \in (u, p']$ *and* $q'' \in (v, p'']$.

Similarly to the previous case, `Routine CriticalTwo()` (in the second part of Routine 1) describes how critical points for pairs can be computed, exploiting Property 6. The correctness of `CriticalOne()` and `CriticalTwo()` follows directly from Properties 5 and 6.

4. Robots' movements. In our algorithm, there are different movement operations, described in the following; in all cases, movements are in straight lines.

Routine 1 Routines for computing critical points.

Routine CriticalOne(u, p)

  Let B be the set returned by Routine 7.2 when executed on $P \setminus \{u\}$.
  For Each $c \in B$, and **For** each gap point x_g for c **Do**
    Add $\overline{up} \cap \uparrow x_g$ to the set of critical points of u.
Routine CriticalTwo($(u', p'), (u'', p'')$)—critical points for u'
  Let B be the set returned by Routine 7.3 when executed on $P \setminus \{u', u''\}$.
  For Each $c \in B$, and **For** each pair of gap points x_g and y_g for c **Do**
    Add $\overline{u'p'} \cap \uparrow x_g$, and $\overline{u'p'} \cap \uparrow y_g$ to the set of critical points of u'.

The basic move. The basic movement operation is moveTo(p), where the robot r moves toward point p; recall that the robot executing this operation might not reach p. If r is inside *SEC*, we denote by moveTo(*SEC*) its movement toward the intersection between *SEC* and the radius where r lies.

To avoid collisions. The operation moveIfFreeWay(p) is used to avoid the creation of unintended dense points and is defined as follows: if no other robot is between robot r and its destination p, then r moves toward p; otherwise, r does not move at all. As a consequence, the following holds.

PROPERTY 8. *If all movements executed in the system from time t to $t' > t$ are due only to* moveIfFreeWay(p) *operations, then during this time no dense points can be created except at p.*

Recall that a robot is *acting on p* at time t if at that time it is moving toward point p or its computed destination is p; we now introduce the following definition.

DEFINITION 4.1 (safely acting). *A robot r is* safely acting *on p if r is acting on p and either* (i) *there is no other robot between r and p, or* (ii) *r moves toward p by executing* moveIfFreeWay(p).

Cautious movements. The next two types of movement, moveCautiously() and movePairwiseCautiously(), are crucial to keep under control when a configuration that is not biangular changes into a biangular configuration because of the movement of one or two robots. In order to introduce them, we exploit the properties of *critical points*, introduced in the previous section.

In the operation moveCautiously(p), a robot first computes the set $\mathcal{CP}(r, p)$ as described in Routine 1. Then, it moves toward the first critical point on its way toward p. As a consequence, we have the following property.

PROPERTY 9. *Let r perform a* moveCautiously(p) *at time t; and let the configuration be* stillBut(r) *from time t to time t' when r stops. Then, at time t'*

 1. *r is at point p, or*
 2. *r is at a point $y \neq p$ closer to p, and where no biangular configuration exists, or*
 3. *r is on a critical point such that the configuration is biangular and still.*

In all cases, no biangular configuration occurs during its movement.

In the following, whenever r is inside a circle $\mathcal{C}$ and p is the intersection of $\mathcal{C}$ and the radius where r lies, we use moveCautiously($\mathcal{C}$) instead of moveCautiously(p).

The operation movePairwiseCautiously($(r', p'), (r'', p'')$) controls the moves of a pair of robots r' and r'' moving toward p' and p'', respectively, with regard to biangularity. The robot r' (similarly, r'') performing this operation first computes all the pairs of critical points and determines those that are in its path to p' (by executing

---

Routine 2 movePairwiseCautiously$((r', p'), (r'', p''))$—executed by r'.

Compute $\mathcal{CP}((r', p'), (r'', p''))$;
$N' :=$ Number of critical points on $\overline{r'p'}$;
$N'' :=$ Number of critical points on $\overline{r''p''}$;
If $N' \neq 0$ **Then** $cp :=$ My next critical point; $hp :=$ Point half way on $\overline{r'cp}$;
Case r', r''
- neither I nor r'' is at a critical point
  Case N', N''
    - $N' = N'' = 1$: moveTo(p').
    - $N' = N'' > 1$: moveTo(cp).
    - $N' > N''$: moveTo(cp).
    - $N' < N''$: do_nothing()
- I am at a critical point and r'' is not
  If $N'' > N'$ **Then** do_nothing() **Else** moveTo(hp).
- only robot r'' is at a critical point and I am not
  Case N', N''
    - $N' = 1 \wedge N'' = 0$: moveTo(p').
    - $N' > 1 \wedge N'' = 0$: moveTo(cp).
    - $N' > N'' > 0$: moveTo(cp).
    - $N'' \geq N'$: do_nothing()
- both I and r'' are at a critical point
  If $N' = 0 \vee N'' > N'$ **Then** do_nothing() **Else** moveTo(hp).

CriticalTwo() in Routine 1). In particular, for all critical points that are between r' and its destination p', we call the *next* critical point the first critical point the robot would reach on its way toward its destination. The robot r' is allowed to move only if it has at least as many critical points ahead as r''. If allowed, it then moves toward either its next critical point or the point half the distance to its next critical point; the choice depends on whether or not r' is already at a critical point when it executes the operation. The pseudocode for operation movePairwiseCautiously$((r', p'), (r'', p''))$ when executed by r' is reported in Routine 2; see also Figure 4.1.

Let $N'(t)$ and $N''(t)$ denote the set of critical points on $\overline{r'p'}$ and on $\overline{r''p''}$, respectively. By definition of movePairwiseCautiously$((r', p'), (r'', p''))$ the next property follows.

PROPERTY 10. *Let r' execute* movePairwiseCautiously$((r', p'), (r'', p''))$ *at time t. Moreover, from time t to the time t' when r' becomes active again, let the configuration be plain and* stillBut$(\{r', r''\})$, *and let r'' be still or executing* movePairwiseCautiously$((r', p'), (r'', p''))$. *Then, r' performs a nonnull movement at time t only if $|N'(t)| \geq |N''(t)|$.*

PROPERTY 11. *Let r' execute* movePairwiseCautiously$((r', p'), (r'', p''))$ *at time t and, as result, perform a nonnull movement. Moreover, from time t to the time t' when r' stops, let the configuration be plain and* stillBut$(\{r', r''\})$, *and let r'' be still or executing the same operation. Then, at time t',*

1. *r' is at point p', or*
2. *r' is at a point $y \neq p'$ closer to p', and where no biangular configuration exists, or*
3. *r' and r'' are on a pair of critical points such that the configuration is biangular, and they are both still.*
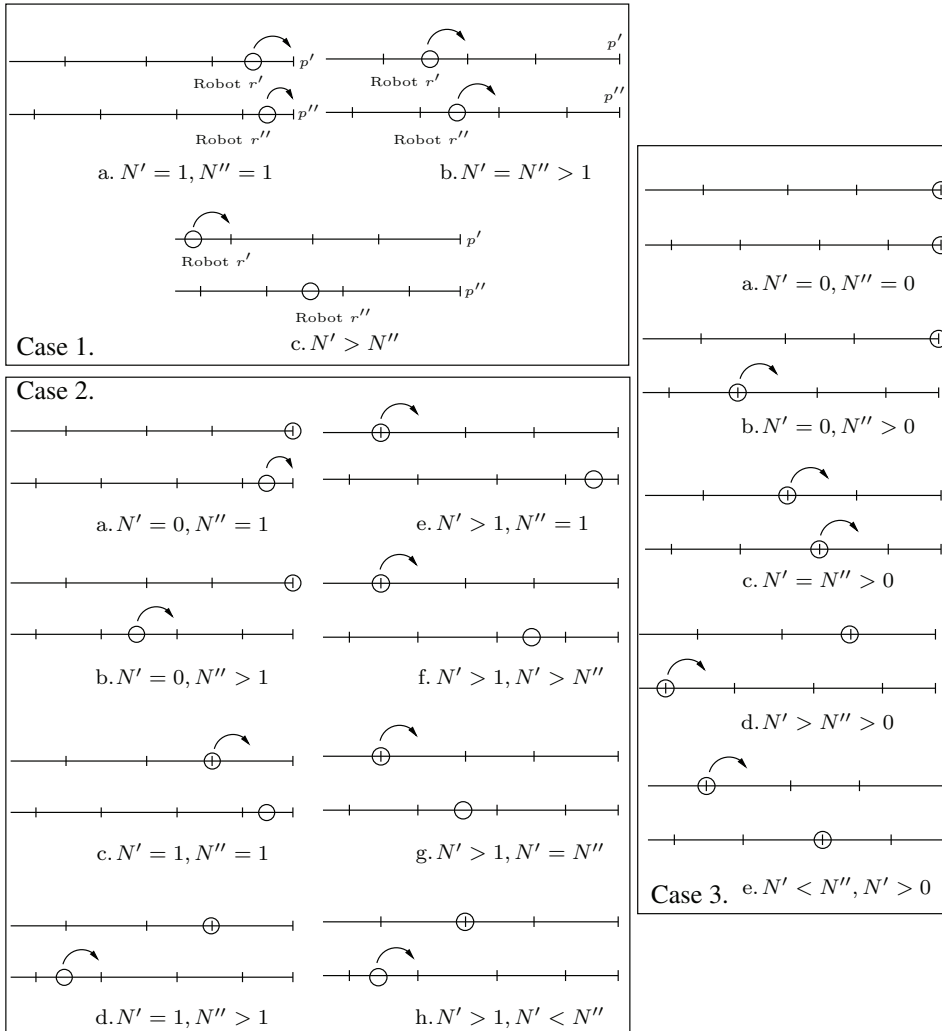
Fig. 4.1. *Case* 1: *Movements if no robot is at a critical point. Case* 2: *Movements if one of the two robots is at a critical point. Case* 3: *Movements if both robots are at a critical point.*

In all cases, no biangular configuration occurs during their movements.

In other words, if r' and r'' are the only robots allowed to move from time t on, and they do so with operation movePairwiseCautiously$((r', p'), (r'', p''))$, they will reach their destination in finite time. Furthermore, the two robots will first reach a situation when $|N'(t')| = |N''(t')|$ (if not already so at time t). From that time on, they will operate in *lock-step*, that is, $abs(|N'(t'')| - |N''(t'')|) \leq 1$ for $t'' \geq t'$, and they will stop at each critical point on their way.

In the following, we shall write movePairwiseCautiously(r', r'', p) instead of movePairwiseCautiously$((r', p), (r'', p))$ to simplify the notation. Also, instead of movePairwiseCautiously$((r', p'), (r'', p''))$, we shall use movePairwiseCautiously$(r', r'', \mathcal{C})$ whenever r' and r'' are inside or on a circle $\mathcal{C}$, and p' and p'' are the intersection of $\mathcal{C}$ and the radius where r' and r'' lie, respectively.
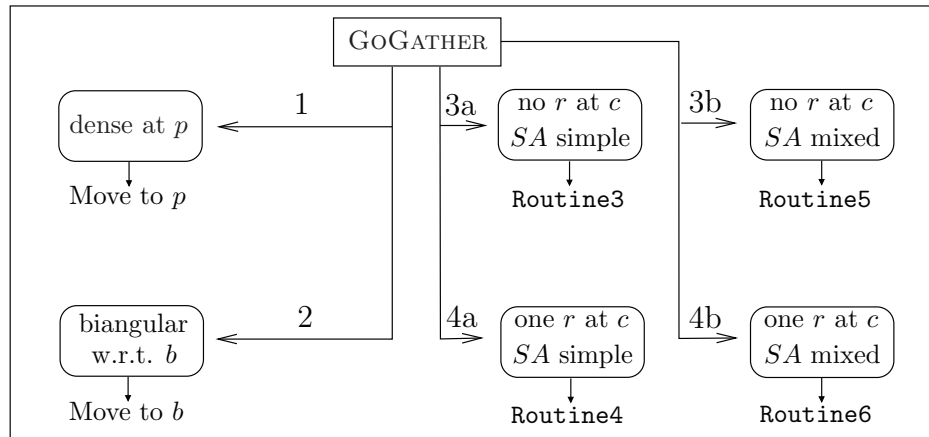
FIG. 5.1. *Schematic overview of our solution; the numbers on the arrows represent the ordering of the tests performed by Algorithm* GoGather.

5. The solution protocol and its correctness. In this section, we describe and analyze the algorithm GoGather. This algorithm is based on a case analysis of the possible initial configurations. We prove that the proposed algorithm solves the Gathering Problem for all initial plain configurations of $n \geq 5$ robots.

5.1. Overall structure. The *global* behavior of our algorithm can be summarized as follows. If the initial configuration is biangular, the robots gather at the center of biangularity. If the initial configuration is nonbiangular, the robots elect a subset of the robots which move and (i) form a dense point at the center of the smallest enclosing circle of all robots, or (ii) form a dense point on the circumference of the smallest enclosing circle of all robots (while such a circle remains invariant), or (iii) form a biangular still configuration, which all robots will eventually notice.

The *local* behavior of the algorithm (i.e., from the point of view of the robot executing it) consists of a sequence of tests on the configuration observed in the *Look* operation; its overall structure is shown in Algorithm 1 (see also Figure 5.1). The robot first determines if there is a dense point, p; if so, the robot moves toward it. Otherwise the robot checks if the observed configuration is biangular; if so, the robot will move toward the center of biangularity, b; as proved later, in this case *all* robots eventually recognize it and move toward b, and within finite time b becomes dense. These two cases are described and analyzed in section 5.2.

If there is no dense point and the configuration is not biangular, the robot examines the string of angles SA of the robots with respect to the center c of the smallest enclosing circle in the observed configuration. The robot determines which of four possible cases the observed configuration falls into; these cases depend on whether there is one or no robot at the center of SEC, and on whether the SA is simple (i.e., the string does not contain any angle of zero degrees) or mixed (i.e., at least one angle of zero degrees is in the string; this implies that at least two robots are on the same radius).

In all four cases, a subset of the robots is elected. A robot is allowed to move only if it is one of the elected robots. In each case, the type of movement used by the elected robots depends on the number of start points in the string of angles and on the number of robots inside SEC (see Figure 5.2). In these movements, particular attention is

payed to avoiding accidental collisions, as well as to the presence of *critical points* (i.e., points where these movements might form biangular configurations). These points are explicitly computed: if an elected robot has one or more critical points on its way, the algorithm ensures that it reaches the first one. If the movements of the elected robots create a biangular configuration, this configuration will eventually be observed by all other robots (and thus they will all eventually gather on the center of biangularity). If no biangular configuration is formed by their movement, the elected robots will create a unique dense point where all other robots will gather: if the elected set consists of a single robot, that robot will move until it reaches another robot, thus creating a single dense point; otherwise, the robots of the elected set move toward the center of the smallest enclosing circle, ensuring that the smallest enclosing circle is not changed by their movements, creating a dense point there.

In particular, the four cases are as follows.

Case 1. If there is no robot at c and $SA()$ is simple (this case is described and analyzed in section 5.3; refer also to Figure 5.2(a)), the behavior of the algorithm depends on the number s of start points of the lexicographically minimum string ($LMS()$). If $s = 1$ and inside SEC there is more than one robot, first all robots inside SEC are sequentially (and cautiously) moved to the rim of SEC, except one; then, this last robot cautiously moves to the center c of the SEC. Otherwise ($s = 1$ and the interior of SEC is empty), one robot is elected as a leader (this is possible since $s = 1$), and this elected robot cautiously moves toward c. After a robot reaches c, the configuration now has one robot at c, and one of Cases 3 or 4 below applies.

If $s = 2$, robots are paired in *teams*; if $s > 2$, robots are grouped into *classes*. In both cases, the algorithm follows an approach similar to $s = 1$: if more than one team or class is inside SEC, first all teams (classes) are sequentially (and cautiously) moved to the rim of SEC, except the robots of one team (class); then, these robots move toward c (the movement is cautious when $s = 2$). Otherwise (the interior of SEC is empty), one team (class) is elected, and the elected robots cautiously move toward c. As a consequence, after a finite time, either two (or more) robots reaches c simultaneously (hence a dense point is created), or one robot reaches c while the others are inside SEC on their way toward c, and Case 2 below applies.

Case 2. If there is one robot at c and $SA()$ is simple (this case is described and analyzed in section 5.5; refer also to Figure 5.2(c)), the algorithm distinguishes on the number NI of robots inside SEC. If $NI = 1$, then the only robot inside SEC is at c; in this case, this robot simply chooses another robot on the rim of SEC and cautiously moves toward it. As soon as the robots leaves c, Case 3 below applies. If $NI = 2$, then the robot inside SEC and not at c moves cautiously toward c; hence, in finite time, a dense point will be created. If $NI > 2$ and the configuration of robots is irregular periodic, then all robots inside SEC move toward c; hence, in finite time, at least two robots will be at c, and a dense point is created. If $NI > 2$ and the configuration of robots is not irregular periodic, the algorithm further distinguishes on the number $\overline{s}$ of start points of the lexicographically minimum string obtained not considering the robot at c. In all these cases, in finite time at least two robots will be at c, and a dense point is created.

Case 3. If there is no robot at c and $SA()$ is mixed (this case is described and analyzed in section 5.4; refer also to Figure 5.2(d)), the algorithm once again distinguishes on s, the number of start points of the lexicographically minimum string. If $s = 1$, a robot can be elected as a leader; furthermore, on the radius where the leader is, there must also be (at least) one other robot: the leader moves cautiously toward
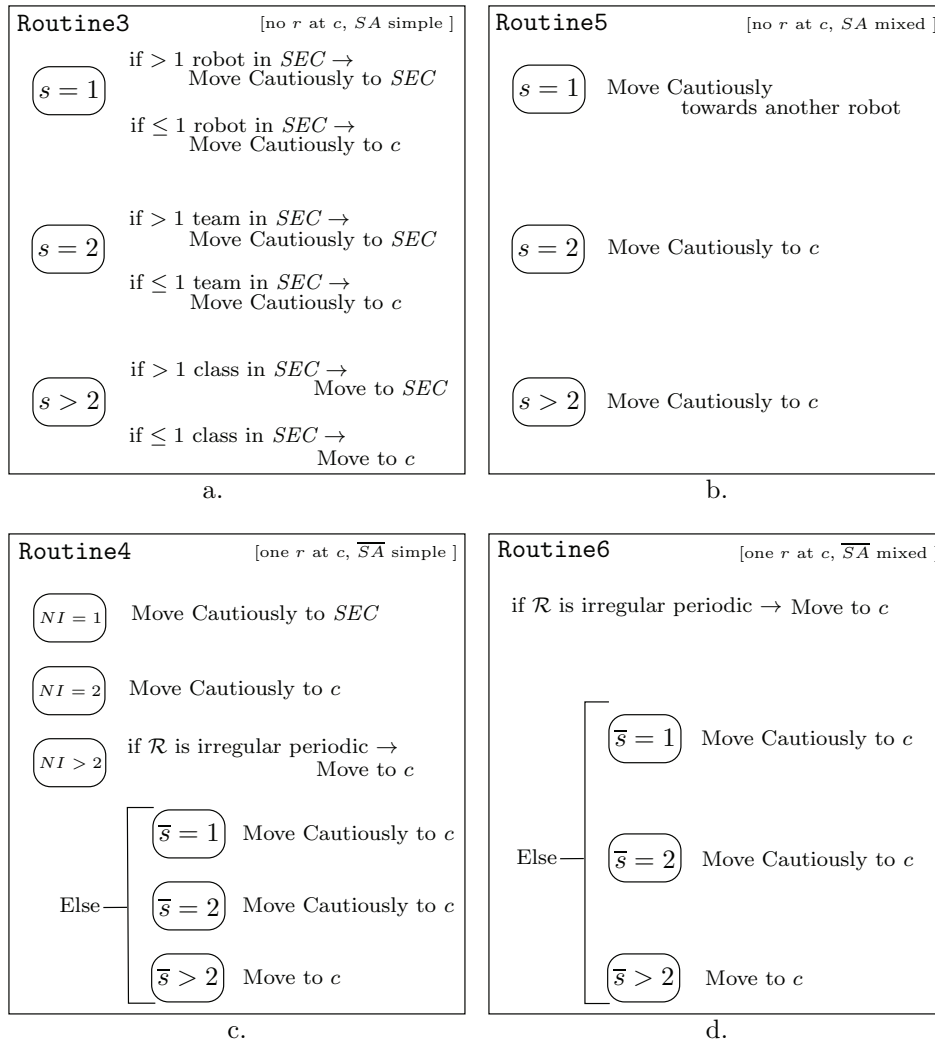
FIG. 5.2. *The four cases of the algorithm when there is no dense point, and the configuration is not biangular. Recall that s is the number of start points in the string of angles; $\overline{s}$ is the number of start points in the string of angles built not considering c (such a string of angles is denoted by $\overline{SA}$ in the figure); and NI is the number of robots inside SEC. Note that the figure specifies only the kind of movement that is performed in each case and not which robot (or subset of robots) is performing it.*

the closest robot on its radius. Hence, in finite time, a dense point is formed.

If $s = 2$, then robots are paired in teams, and one team can be elected as leader; the robots in the elected team move cautiously toward c. Finally, if $s > 2$, the robots are grouped into classes, and one class is elected as leader. The robots in the elected class move toward c. Hence, in both these cases, in finite time a dense point is created at c.

Case 4. If there is one robot at c and $SA()$ is mixed (this case is described and analyzed in section 5.6; refer also to Figure 5.2(e)), if the configuration of robots is irregular periodic, then the robots can be grouped into classes, and one class is elected as a leader: the robots from this class move toward c.
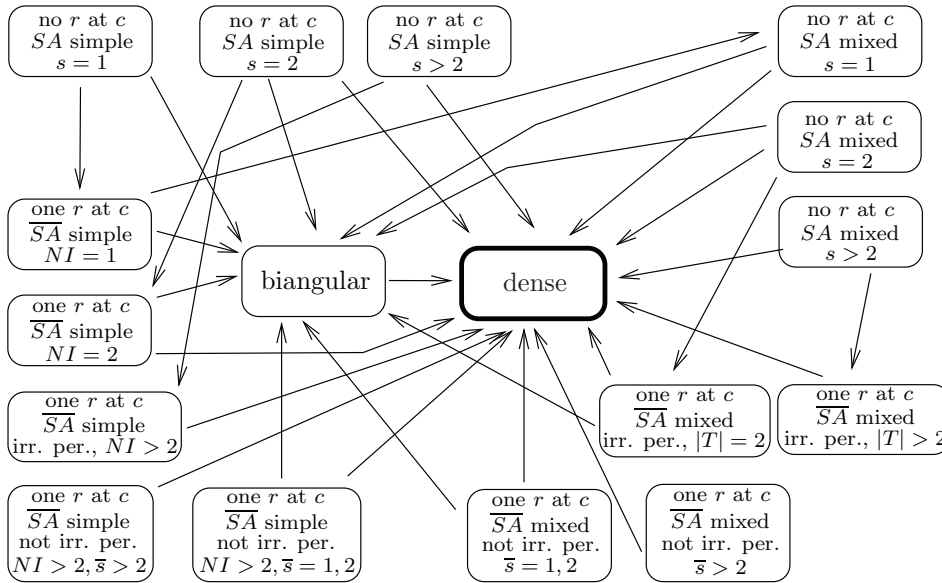
FIG. 5.3. *The possible transitions among the configurations;* $\overline{SA}$ *and* $\overline{s}$ *are as in Figure* 5.2.

If the configuration is not irregular periodic, the algorithm distinguishes on the number $\overline{s}$ of start points of the lexicographically minimum string obtained not considering the robot at c and behaves similarly to Case 2, when $NI > 2$. In both cases, in finite time a dense point is formed at c.

The possible transitions between configurations induced by the execution of the algorithm are shown in Figure 5.3.

5.2. First tests. The first test a robot does when computing is to determine whether there is a single dense point, p; if so, all robots will gather there, carefully avoiding collisions, i.e., the (even temporary) creation of another dense point. Recall that, by definition, the initial configuration of the robots is plain (i.e., there is no dense point); as we will prove, our algorithm ensures that exactly one dense point is created.

THEOREM 5.1. *Let p be the only dense point at time t. If at that time all the robots are either still or safely acting on p, then there exists a time $t^* > t$ when all robots gather at p.*

Proof. First we show that, in finite time, a robot either reaches p or moves exclusively according to moveIfFreeWay(p). Let $\mathcal{A}(t)$ be the set of robots that are active at time t. Any robot in $\mathcal{A}(t)$ is, by hypothesis, safely acting on p; hence, by Property 8 and Definition 4.1, no collisions can occur during these movements; that is, no other point becomes dense even temporarily. Also, each robot in $\mathcal{A}(t)$, during the cycle it is executing at time t, either reaches p or stops before p. In the latter case, in the next cycle it executes, according to Algorithm GOGATHER, it will move toward p according to operation moveIfFreeWay(p). Furthermore, since the active robots in $\mathcal{A}(t)$ do not create a dense point other than p, any robot that is not active at t, the next time it becomes active, if not yet at p, will, by Algorithm GOGATHER, move toward p according to operation moveIfFreeWay(p).

Algorithm 1 GoGather.

$\mathcal{R}$:= Set of positions of the robots;
If One dense point p **Then** moveIfFreeWay(p).
Else
  If The robots are in regular (resp., irregular) biangular configuration **Then**
5:    b := Center of regular (resp., irregular) biangularity;
    moveIfFreeWay(b).
  Else
    SEC := Smallest Enclosing Circle of all robots;
    c := Center of SEC;
10:    **If** No robot is at c **Then**
      Compute the set of strings $SA(\mathcal{R})$, $LMS(\mathcal{R})$;
      Compute $StS^+(\mathcal{R})$, $StS^-(\mathcal{R})$;
      $s := |StS^+(\mathcal{R}) \cup StS^-(\mathcal{R})|$;
      If $SA(\mathcal{R})$ is simple **Then** Routine3. **Else** Routine5.
15:    **Else** %One robot r is at c%
      $\overline{\mathcal{R}} := \mathcal{R} \setminus \{c\}$;
      Compute the set of strings $SA(\overline{\mathcal{R}})$, $LMS(\overline{\mathcal{R}})$;
      Compute $StS^+(\overline{\mathcal{R}})$, $StS^-(\overline{\mathcal{R}})$;
      $s := |StS^+(\overline{\mathcal{R}}) \cup StS^-(\overline{\mathcal{R}})|$;
20:      **If** $SA(\overline{\mathcal{R}})$ is simple **Then** Routine4. **Else** Routine6.

Therefore, in finite time, each robot either is at p or moves exclusively according to moveIfFreeWay(p). Hence, by definition of moveIfFreeWay(p), by Property 8, and by Assumption Dis, within finite time all robots gather at p, and the theorem follows. □

In the absence of a dense point, the next check is whether the configuration is biangular (regular or irregular). Should this be the case, the robots will gather at the center of biangularity.

THEOREM 5.2. *Let at time t the configuration be plain, still, and biangular (either regular or irregular) with center b. Then there exists a time $t^* > t$ when all robots gather at b.*

Proof. First observe that, if the movements performed from time t on are only according to operation moveIfFreeWay(b), the configuration remains biangular, and, by Property 8, no collisions can occur at points other than b.

Thus, if the configuration is plain, still, and regular biangular with center b at time t, when a robot becomes active, according to Algorithm GoGather, it can only execute moveIfFreeWay(b); hence, by Assumption Dis, within finite time one or more robots will reach b. Let t' be the first time b is reached; if two or more robots reach b at t', b becomes dense, and all active robots not on p are safely acting on p; thus, by Theorem 5.1, all robots will gather at b within finite time. If only one robot reaches b at time t', the configuration becomes biangular irregular. This means that while some robots might be safely acting on b because of earlier observations, others might now act based on the irregularity of the biangular configuration. According to the algorithm, the latter robots will safely act on the center b' of irregular biangularity by executing moveIfFreeWay(b'). However, by definition of biangular irregular configuration, $b' = b$; hence, by Assumption Dis and Property 8, within finite time b will become dense, and all robots will gather there by Theorem 5.1.

Similarly, if at time t the configuration is plain, still, and irregular biangular with center b (by Lemma 2.6 the center is unique), the robot already at b stays still. According to the algorithm, all other robots can only execute `moveIfFreeWay(b)`; hence, within finite time, b will become dense, with all acting robots safely acting on b. Therefore, by Theorem 5.1, all robots will gather there within finite time. □

5.3. No robot at the center of SEC and SA is simple. The next test a robot performs is whether no robot is at the center of SEC and SA is simple. Let c bet the center of $SEC(\mathcal{R})$. Consider the number s of starting positions of $LMS(\mathcal{R}, c)$, i.e., the cardinality of the set $StS^+(\mathcal{R}, c) \cup StS^-(\mathcal{R}, c)$. In the following, when no ambiguity arises, we will use the notation $(\mathcal{R})$ instead of $(\mathcal{R}, c)$.

The overall structure of the algorithm in this case is to first carefully move all robots that might be inside SEC on the circumference of SEC (with some exceptions). Then, some robots are elected as *leaders* (their number depends on the value of s); finally, the leaders are moved toward the center of SEC. Depending on the value of s, we distinguish three possible cases.

5.3.1. One starting position of $LMS(\mathcal{R})$ ($s = 1$). In case there is a unique starting position r_k for $LMS(\mathcal{R})$, by Lemma 2.1, $SA^+(\mathcal{R})[k] \neq SA^-(\mathcal{R})[k]$; without loss of generality, let $LMS(\mathcal{R}) = SA^+(\mathcal{R})[k]$. We use the ordering of the robots in $SA^+(\mathcal{R})[k]$ to achieve different means.

We first consider the two cases when there is only one robot inside SEC (e.g., as a result of the scenario considered by the above lemma), and when all robots are on SEC (e.g., as an initial configuration). In both cases, we use the total ordering implied by $s = 1$ to elect a robot as the *leader*, using operation `ElectOne()` defined as follows. If there is only one robot inside SEC, this robot is chosen as the leader. If all robots are on SEC, then the leader is the first robot l (according to the ordering) such that SEC remains invariant if l is removed; note that, since $n \geq 5$, by Property 1 such a robot exists. Once the leader is elected, it moves cautiously toward the center of SEC. Hence we have following lemma.

LEMMA 5.3. *Let the configuration at time t be plain and still, no robot be at c, $SA(\mathcal{R})$ be simple, $s = 1$, and at most one robot be inside SEC. Then there exists a time $t^* > t$ when the robots reach a plain and still configuration $\mathcal{H}$ such that either*

1. *$\mathcal{H}$ is biangular, or*
2. *in $\mathcal{H}$ only one robot is inside SEC and at c, and $SA(\mathcal{R} \setminus \{c\})$ is simple.*

Proof. According to `Routine3`, a single robot, say, l, is elected as leader by `ElectOne()`. The move operation l performs is `moveCautiously(c)`; moreover, all other robots stay still during this movement.

By Property 9, at time t' when l stops, either l reaches c, or a biangular configuration is formed, or l is the only robot inside SEC, and its distance to c has decreased. Note that, in all cases, at time t' the configuration is still and plain. In the first case, $SA(\mathcal{R} \setminus \{c\})$ is simple. In the last two cases, $SA(\mathcal{R})$ does not change during the movement; hence, it is still *simple* and $s = 1$. Thus, in the first two cases, the lemma holds. In the last case, the hypotheses of the lemma are still met with l closer to c; hence, by Assumption DIS, the lemma will eventually hold. □

If there is more than one robot inside SEC, we use the ordering to sequentially move to SEC each of those robots but one. Each of these robots is selected in turn, according to the ordering, in an operation called `SelectOneInside()`; more precisely, this operation selects the first robot (according to the total ordering) that is inside SEC. The chosen robot moves cautiously toward SEC, while all others stay still; this

---

Routine 3 Subroutine: No robot at c, SA is simple.

Case s
- $s = 1$
  - **If** There is more than one robots inside SEC **Then**
    - l:=SelectOneInside();
- 5:   **If** I am l **Then** $p :=$ Intersection between SEC and $Rad(l)$; moveCautiously(p).
  - **Else** %Either SEC is empty, or there is only one robot inside%
    - $l :=$ElectOne();
    - **If** I am l **Then** moveCautiously(c)
- $s = 2$
- 10:   **If** There are robots inside SEC from more than one team **Then**
  - $T :=$SelectTeamInside();
  - **If** $|T| = 1$ **Then**
    - $l :=$ Robot in T;
    - **If** I am l **Then** moveCautiously(SEC).
- 15:   **Else** %Two robots in T%
  - $(l', l'') :=$ Robots in T;
  - **If** I am l' or l'' **Then** movePairwiseCautiously(l', l'', SEC)
  - **Else** %Either SEC is empty, or only robots from the same team are inside SEC%
    - $T :=$ElectTeam();
- 20:   **If** I am in T and $|T| = 1$ **Then** moveCautiously(c).
  - **Else** %Two robots in T%
    - **If** I am in T **Then**
      - $r :=$ Point where I am; $p :=$ Half point on $Rad(r)$;
      - $r' :=$ Point where my teammate is; $p' :=$ Half point on $Rad(r')$;
- 25:   **If** Both robots of T are inside SEC **Then**
  - movePairwiseCautiously(r, r', c).
  - **If** Both robots of T are on SEC and $|\mathcal{CP}((r,c),(r',c))| = 1$ **Then** moveTo(p).
  - **If** Both robots of T are on SEC and $|\mathcal{CP}((r,c),(r',c))| > 1$ **Then** movePairwiseCautiously(r, r', c).
- 30:   **If** I am the only robot of T on SEC and $|\mathcal{CP}((r,c),(r',c))| = 1$ **Then** moveTo(p).
  - **If** I am the only robot of T on SEC and $|\mathcal{CP}((r,c),(r',c))| > 1$ **Then** movePairwiseCautiously(r, r', c).
  - **If** I am the only robot of T inside SEC and $|\mathcal{CP}((r,c),(r',c))| > 1$ **Then** movePairwiseCautiously(r, r', c).
- 35: $s > 2$
  - **If** MoreOneClassInside() **Then**
    - SC:=SelectClassInside();
    - **If** I am in SC and inside SEC **Then** moveTo(SEC)
  - **Else** %Either SEC is empty, or inside there are only robots from the same class%
- 40: T:=ElectClass();
  - **If** I am in T **Then**
    - $NI :=$ Number of robots inside SEC;
    - **If** $|NI| = 0$ **Then**
      - **If** I am in T **Then** moveTo(half point on my radius).
- 45:   **If** $|NI| = 1$ **Then**
  - **If** I am in T **Then** do_nothing().
  - **If** $|NI| \geq 2$ **Then**
    - **If** I am in T **Then** moveTo(c).

---

operation is repeated until either a biangular configuration has been formed or all robots except one are on *SEC*.

LEMMA 5.4. *Let the configuration at time t be plain and still, no robot be at c, $SA(\mathcal{R})$ be simple, $s = 1$, and more than one robot be inside SEC. Then there exists a time $t^* > t$ when the robots reach a plain and still configuration $\mathcal{H}$ such that either*

1. *$\mathcal{H}$ is biangular, or*
2. *in $\mathcal{H}$ no robot is at c, $SA(\mathcal{R})$ is simple, $s = 1$, and only one robot is inside SEC.*

Proof. We proceed by induction on the number $m \geq 1$ of robots inside *SEC*. The lemma trivially holds for $m = 1$. Let it hold for $1 \leq m \leq k$ robots, and consider the case when inside *SEC* there are $k + 1$ robots. According to `Routine3`, only one robot inside *SEC* is allowed to move—the one chosen by `SelectOneInside()`. In this case, the move operation r performs is `moveCautiously(`p`)`, where p is the intersection between *SEC* and $Rad(r)$; moreover, all other robots stay still during this movement. By Property 9, at time t' when r stops, (i) a biangular configuration is formed, or (ii) r reaches *SEC*, decreasing by one the number of robots inside *SEC*, or (iii) r is still inside *SEC*, and its distance to it has decreased. Note that, in all cases, $SA(\mathcal{R})$ does not change during the movement; hence, it is still *simple* and $s = 1$. Furthermore, at time t' the configuration is still and plain. Thus, in case (i) the lemma holds. In case (ii) the lemma holds by induction. In case (iii) the hypotheses of the lemma are still met with r closer to *SEC*; hence, by Assumption DIS, conditions (i) or (ii), and thus the lemma, will eventually hold. $\square$

As a consequence, if a biangular configuration is not formed, within finite time there will be exactly one robot inside *SEC* (and Lemma 5.3 applies). By Lemmas 5.3 and 5.4 we can state the following theorem.

THEOREM 5.5. *Let the configuration at time t be plain and still, no robot be at c, $SA(\mathcal{R})$ be simple, $s = 1$. Then there exists a time $t^* > t$ when the robots reach a plain and still configuration $\mathcal{H}$ such that either*

1. *$\mathcal{H}$ is biangular, or*
2. *in $\mathcal{H}$ only one robot is inside SEC and at c, and $SA(\mathcal{R} \setminus \{c\})$ is simple.*

5.3.2. Two starting positions of $LMS(\mathcal{R})$ ($s = 2$). In this case, we can pair robots in teams of at most two elements and totally order the teams as described in section 2.2.2. We consider first the two cases when all robots are on *SEC*, and when there are only robots from one team inside *SEC*. In both cases, we use the total order of the teams to elect a team as the *leader*, using operation `ElectTeam()` defined as follows. If there is only one team inside *SEC*, this team is chosen as the leader. If all robots are on *SEC*, then the leader is the first team T (according to the ordering) such that *SEC* remains invariant if robots in T are removed. Note that, since $n \geq 5$, such a team exists, as shown by Lemma 2.2.

Once the leader team is elected, robots from this team move cautiously toward the center of *SEC*. Note that the algorithm ensures that both robots from the team are inside *SEC* before allowing one of them to reach the center c. If the robots have more than one critical point toward their path to c, this is implicitly ensured by `movePairwiseCautiously()`; otherwise, the robots are first moved toward the median point on their radius.

LEMMA 5.6. *Let the configuration at time t be plain and still, $SA(\mathcal{R})$ be simple, $s = 2$, and all robots be on SEC. Then there exists a time $t^* > t$ when the robots reach a configuration $\mathcal{H}$ such that either*

1. *$\mathcal{H}$ is biangular and still, or*

2. $\mathcal{H}$ *is plain; only robots from one team* $T = \{l', l''\}$ *are inside SEC;* $\mathcal{H}$ *is still, or* stillBut(l) *with* $l \in T$ *either acting on the median point on* $Rad(l)$ *and* $|\mathcal{CP}((l', c), (l'', c))| = 1$, *or performing* movePairwiseCautiously(l', l'', c). *Furthermore, in* $\mathcal{H}$ *both robots in* T *are inside SEC but not at* c, *and* $s = 2$.

Proof. If all robots are on *SEC*, by Lemma 2.2, a unique team T of two robots is elected by ElectTeam(); let $T = \{r_k, r_w\}$. Let p_k and p_w be the half points between their positions and c, respectively, at time t. Let $N'(t)$ and $N''(t)$ be the number of critical points of r_k and r_w at time t, respectively; without loss of generality, let $N'(t) \geq N''(t)$. We distinguish the possible cases.

1. If $N'(t) = N''(t) = 1$ (i.e., c is the only critical point for both robots), the algorithm forces r_k and r_w to perform moveTo(p_k) and moveTo(p_w), respectively. Note that, if one of the two robots stops inside *SEC* while the other is still on *SEC*, it will wait. Thus, by Assumption DIS, within finite time, they will both be inside *SEC*, with at least one of them still, and the lemma follows.

2. If $N'(t) > N''(t) = 1$, then the algorithm forces r_k and r_w to perform move-PairwiseCautiously(r_k, r_w, c). By definition of movePairwiseCautiously(), by Properties 10 and 11, and by Routine3, r_k is the only one allowed to move inside *SEC* as long as no regular biangular configuration is formed (r_w can be on a critical point on *SEC*) and $N' > 1$. By Assumption DIS, within finite time, say, at t', $N'(t') = 1$. At this time Routine3 forces r_k to not move (at time t', r_w is on *SEC*) and forces r_w to move toward p_w. The lemma follows as soon as r_w moves inside *SEC*.

3. If $N'(t) = N''(t) > 1$, then the algorithm forces r_k and r_w to perform movePairwiseCautiously(r_k, r_w, c); then, by Properties 10 and 11, the two robots are forced to move toward c in lock-step (both executing movePairwiseCautiously(r_k, r_w, c)), and the lemma follows.

4. If $N'(t) > N''(t) > 1$, by definition of movePairwiseCautiously(), by Properties 10 and 11, and by Routine3, r_k is the only one allowed to move inside *SEC* as long as no regular biangular configuration is formed (r_w can be on a critical point on *SEC*) and $N' \neq N''$. By Assumption DIS, within finite time, say, at t', $N'(t') = N''(t')$. At this time Routine3 forces r_k to not move (at time t', r_w is on the *SEC*) and forces r_w to move toward c by invoking movePairwiseCautiously(r_k, r_w, c). The lemma follows as soon as r_w moves inside *SEC*.

Note that, in all the above arguments, the movements of both robots can only be along $Rad(r_k)$ and $Rad(r_w)$, respectively; hence, during these movements the configuration remains plain, $SA(\mathcal{R})$ does not change (it stays *simple*), $s = 2$, and the robots in T are the only ones allowed to move. $\quad\square$

LEMMA 5.7. *Let the configuration at time* t *be plain, no robot be at* c, $SA(\mathcal{R})$ *be simple,* $s = 2$, *only robots of one team* T *be inside SEC, with* $T = \{l', l''\}$, *and at least one robot of* T *be inside SEC. Furthermore, let the configuration be still, or* stillBut(l), $l \in T$, *with* l *either acting on the median point on* $Rad(l)$ *and* $|\mathcal{CP}((l', c), (l'', c))| = 1$, *or performing* movePairwiseCautiously(l', l'', c). *Then there exists a time* $t^* > t$ *when the robots reach a configuration* $\mathcal{H}$ *such that*

1. $\mathcal{H}$ *is biangular and still, or*
2. $\mathcal{H}$ *is dense at* c *and still, or*
3. *in* $\mathcal{H}$ *one robot of the team, say,* l', *is at* c; $SA(\mathcal{R} \setminus \{c\})$ *is simple;* l'' *is the only other robot inside SEC; and* $\mathcal{H}$ *is plain and still or* stillBut(l''), *with*

l'' *safely acting on* c.

Proof. By definition, T is the team selected by `ElectTeam()`. We consider the two possible cases.

Case 1. Consider first the case when both robots in T are inside SEC. Observe the following (whose proof can be found in Appendix C).

CLAIM 1. *Within finite time one of the following conditions holds:*

(i) l' *and* l'' *are on a pair of critical points such that the configuration is biangular, and they are both still.*

(ii) l' *and* l'' *are both on* c, *and they are both still.*

(iii) l'' *is on* c, *and* l' *is inside* SEC.

(iv) *no robot is on* c, *and* l'' *is at a point closer to* c *where no biangular configuration exists.*

If condition (i) or (ii) occurs, the lemma holds. In case condition (iii) occurs, $SA(\mathcal{R} \setminus \{c\})$ is simple. Moreover, l' is either still or is moving toward c executing operation `movePairwiseCautiously`(l', l'', c): in both cases the lemma holds. In the last case, when condition (iv) occurs, l' is again either still or performing `movePair-wiseCautiously`(l', l'', c); by the lock-step movements implied by Property 10, one of the conditions (i), (ii), or (iii) will eventually hold, with l'' closer to c. Hence, by Assumption DIS, the lemma will eventually hold.

Case 2. The second case is when only one robot of the team is inside SEC at time t. Without loss of generality, let l' be the one inside SEC. First observe the following claim (whose proof can be found in Appendix C).

CLAIM 2. *Within finite time either both robots in* T *are inside* SEC, *or a still biangular configuration is formed.*

Thus, the lemma holds either directly or by Case 1. ☐

Now, we consider the case when only a single team T is inside SEC, with $|T| = 1$.

LEMMA 5.8. *Let the configuration at time* t *be plain and still, no robot be at* c, $SA(\mathcal{R})$ *be simple,* $s = 2$, *and exactly one team* T *be inside* SEC, *with* $|T| = 1$. *Then there exists a time* $t^* > t$ *when the robots reach a plain and still configuration* $\mathcal{H}$ *such that either*

1. $\mathcal{H}$ *is biangular, or*
2. *in* $\mathcal{H}$ *one robot is at* c, $SA(\mathcal{R} \setminus \{c\})$ *is simple, and no other robot is inside* SEC.

Proof. By `Routine3`, the only team inside SEC is elected by `ElectTeam()`. The move operation that the robot in T, say, l, performs is `moveCautiously`(c); moreover, all other robots stay still during this movement.

By Property 9, at time t' when l stops, l reaches c, or a biangular configuration is formed, or l is the only robot inside SEC, and its distance to c has decreased. Note that, in all cases, at time t' the configuration is still and plain. In the first case, $SA(\mathcal{R} \setminus \{c\})$ is simple. In the last two cases, $SA(\mathcal{R})$ does not change during the movement; hence, it is still *simple* and $s = 2$. Thus, in the first two cases, the lemma holds. In the last case, the hypotheses of the lemma are still met with l closer to c; hence, by Assumption DIS, the lemma will eventually hold. ☐

If there are robots from more than one team inside SEC, we use the total ordering to sequentially move to SEC every team but one until we reach a situation considered in Lemma 5.7 or 5.8. Each of the teams inside SEC is selected in turn, according to the total ordering, in an operation called `SelectTeamInside()`; the team moves cautiously toward SEC.

LEMMA 5.9. *Let the configuration at time* t *be plain, no robot be at* c, $SA(\mathcal{R})$

be simple, $s = 2$, and robots from more than one team be inside SEC; furthermore, let the configuration be still, or stillBut(l), *with $l \in T$, where T is the smallest team inside SEC at time t, and l performing* movePairwiseCautiously(l', l'', SEC) *(if $T = \{l', l''\}$) or* moveCautiously(SEC) *(if $T = \{l\}$). Then there exists a time $t^* > t$ when the robots reach a plain and still configuration $\mathcal{H}$ such that either*

1. *$\mathcal{H}$ is biangular, or*
2. *in $\mathcal{H}$ no robot is at c, $SA(\mathcal{R})$ is simple, $s = 2$, and only robots from one team are inside SEC.*

Proof. We proceed by induction on the number $m \geq 1$ of teams where at least one robot is inside *SEC*. The lemma trivially holds for $m = 1$. Let it hold for $1 \leq m \leq k$ teams, and consider the case when inside *SEC* there are robots from $k + 1$ teams. According to Routine3, only one team T is allowed to move—the one chosen by SelectTeamInside(). We distinguish the three possible cases.

Case 1. $T = \{r\}$. In this case, r performs moveCautiously(SEC); moreover, all other robots stay still during this movement. By Property 9, at time t' when r stops, (i) a biangular configuration is formed, or (ii) r reaches *SEC*, decreasing by one the number of teams inside *SEC*, or (iii) r is still inside *SEC*, and its distance to it has decreased. Note that, in all cases, $SA(\mathcal{R})$ does not change during the movement; hence, it is still *simple* and $s = 2$. Furthermore, at time t' the configuration is still and plain. Thus, in case (i) the lemma holds. In case (ii), by inductive hypotheses the lemma holds. In the last case, the hypotheses of the lemma are still met with r closer to *SEC*; hence, by Assumption DIS, condition (i) or (ii) will eventually hold. Thus the lemma will eventually hold either directly (in case (i)) or by inductive hypothesis (case (ii)).

Case 2. $T = \{r', r''\}$, and one of them, say, r'', is on *SEC*. By Routine3, the move operation they perform is movePairwiseCautiously(r', r'', SEC). In this case, r'' will perform only null moves (and hence be still) as long as r' is inside *SEC*. Let t' be the first time when r' stops. By Property 11, at time t', (i) a regular biangular configuration is formed, or (ii) r' reaches *SEC*, decreasing by one the number of teams inside *SEC*, or (iii) r' is still inside *SEC* and its distance to it has decreased. Note that, in all cases, $SA(\mathcal{R})$ does not change during the movement; hence, it is still *simple* and $s = 2$. Furthermore at time t' the configuration is still and plain. Thus, in case (i) the lemma holds. In case (ii), by inductive hypothesis, the lemma holds. In the last case, the hypotheses of the lemma are still met with r' closer to *SEC*; hence, by Assumption DIS, condition (i) or (ii) will eventually hold. Thus the lemma will eventually hold either directly (in case (i)) or by inductive hypothesis (case (ii)).

Case 3. $T = \{r', r''\}$, and both robots are inside *SEC*. By Routine3, the move operation they perform is movePairwiseCautiously(r', r'', SEC). Without loss of generality, let r' be the first to execute the operation with a nonnull movement, and let t' be the first time when r' stops. During r''s movement only r'' is allowed to move, by performing movePairwiseCautiously(r', r'', SEC). Since the movements of both robots are only along $Rad(r')$ and $Rad(r'')$, respectively, during these movements the configuration remains plain and $SA(\mathcal{R})$ does not change; hence, it is still *simple* and $s = 2$. This is also true if only r' moves during this time. By Property 11, at time t' when r' stops, (i) r' and r'' are on a pair of critical points such that the configuration is biangular, and they are both still, or (ii) r' is on *SEC*, or (iii) r' is at a point closer to *SEC* and where no biangular configuration exists. In all three cases, at time t', robot r'' is either still or performing movePairwiseCautiously(r', r'', SEC). In case (i) the lemma holds. In case (ii), if r'' is inside *SEC*, we are in Case 2 above, and

hence the lemma holds; otherwise there are only k teams now inside SEC, and the lemma holds by inductive hypothesis. In case (iii), the hypotheses of the lemma are still met, and the distance of the team from SEC (defined as the sum of the distances of r' and r'' from SEC) has decreased. Hence, by Assumption DIS, condition (i) or (ii) will eventually hold. Thus the lemma will eventually hold either directly (in case (i)) or by inductive hypothesis (case (ii)).   □

Summarizing, by Lemmas 5.6–5.9, we have the following.

LEMMA 5.10. *Let the configuration at time t be plain and still, no robot be at c, $SA(\mathcal{R})$ be simple, and $s = 2$. Then there exists a time $t^* > t$ when the robots reach a configuration $\mathcal{H}$ such that*

1. *$\mathcal{H}$ is biangular and still, or*
2. *$\mathcal{H}$ is dense at c and still, or*
3. *$\mathcal{H}$ is plain with one robot at c; $SA(\mathcal{R} \setminus \{c\})$ is simple; at most one other robot r is inside SEC, with r in the same team of the robot at c; and $\mathcal{H}$ is either still or $\mathtt{stillBut}(r)$, with r safely acting on c.*

5.3.3. Many starting positions of $LMS(\mathcal{R})$ ($s > 2$). Let there be more than two starting positions of $LMS(\mathcal{R})$ in $SA^+(\mathcal{R}, c)$ and $SA^-(\mathcal{R}, c)$. The behavior of the algorithm in this case is similar to the previous case $s = 2$; the only difference is that in this case the movements of the robots do not need to be cautious because of Lemma 2.5. The proofs of the lemmas in this section follow the same reasoning of the corresponding lemmas of section 5.3.2, and they are reported in Appendix D.

We consider first the cases when all robots are on SEC, and when there are only robots from one class inside SEC. In both cases, we use the ranking of the teams implied by $s > 2$ to elect a class as the *leader*, using operation $\mathtt{ElectClass}()$ defined as follows: if inside SEC there are only robots from the same class, this class is chosen as the leader; if all robots are on SEC, then the leader is the first class T according to the ordering.

LEMMA 5.11. *Let the configuration at time t be plain and still, $SA(\mathcal{R})$ be simple, $s > 2$, and all robots be on SEC. Then there exists a time $t^* > t$ when the robots reach a configuration $\mathcal{H}$ such that*

1. *only robots from one equivalence class T are inside SEC;*
2. *at least two robots are inside SEC and no robot is at c; and*
3. *$\mathcal{H}$ is plain with $s > 2$; it is still, or $\mathtt{stillBut}(T')$, with $T' \subseteq T$, and where each robot is safely acting on either c or the median point of its radius.*

LEMMA 5.12. *Let the configuration at time t be plain, no robot be at c, at least two robots be inside SEC, and all robots inside SEC be from the same class T, $SA(\mathcal{R})$ be simple, and $s > 2$. Furthermore, let the configuration be still, or $\mathtt{stillBut}(T')$, with $T' \subseteq T$, where each acting robots is safely acting on either c or the median point of its radius. Then there exists a time $t^* > t$ when the robots reach a configuration $\mathcal{H}$ such that either*

1. *$\mathcal{H}$ is dense at c and all acting robots are safely acting on c, or*
2. *$SA(\mathcal{R} \setminus \{c\})$ is simple and $\mathcal{R}$ is irregular periodic; inside SEC there are at least two robots, they are all from T, and one of them, say, r, is at c; and $\mathcal{H}$ is plain, still or $\mathtt{stillBut}(T'')$, with $T'' \subseteq T \setminus \{r\}$, and with all robots in T'' safely acting on c.*

If there are robots from more than one class inside SEC, we use the total ordering to sequentially move to SEC every class but one, until we reach a situation considered in Lemma 5.12. Each of the classes inside SEC is selected in turn, according to the total ordering, in an operation called $\mathtt{SelectClassInside}()$; the robots in the

selected class move toward SEC.

LEMMA 5.13. *Let the configuration at time t be plain, no robot be at c, $SA(\mathcal{R})$ be simple, $s > 2$, and robots from more than one class be inside SEC; furthermore, let the configuration be still, or* stillBut(T'), *where $T' \subseteq T$, with T the smallest class inside SEC at time t, and robots in T' performing* moveTo(SEC). *Then there exists a time $t^* > t$ when the robots reach a plain and still configuration $\mathcal{H}$ such that in $\mathcal{H}$ no robot is at c, $SA(\mathcal{R})$ is simple, $s > 2$, and only robots from one class are inside SEC.*

Summarizing, by Lemmas 5.11–5.13, we have the following.

LEMMA 5.14. *Let the configuration at time t be plain and still, no robot be at c, $SA(\mathcal{R})$ be simple, and $s > 2$. Then there exists a time $t^* > t$ when the robots reach a configuration $\mathcal{H}$ such that either*

1. $\mathcal{H}$ *is dense at c and all acting robots are safely acting on c, or*
2. $SA(\mathcal{R} \setminus \{c\})$ *is simple and $\mathcal{R}$ is irregular periodic; inside SEC there are at least two robots, they are all from the same class T, and one of them, say, r, is at c; and $\mathcal{H}$ is plain, still or* stillBut(T'), *with $T' \subseteq T \setminus \{r\}$, and with all robots in T' safely acting on c.*

5.4. A robot at the center of SEC and $SA(\mathcal{R} \setminus \{c\})$ is simple. Let us now consider the case when there is exactly one robot r at the center c of SEC, and $SA(\mathcal{R} \setminus \{c\})$ (the string of angles of all robots except r) is simple. Let $\overline{s}$ be the cardinality of $StS^+(\mathcal{R} \setminus \{c\}) \cup StS^-(\mathcal{R} \setminus \{c\})$ (therefore, $\overline{s}$ denotes the number of starting positions of $LMS(\mathcal{R} \setminus \{r\}, c)$).

In this case, the algorithm (reported in Routine4) distinguishes three possible cases, according to the number NI of robots that are inside SEC.

If r is the only robot inside SEC, then r chooses an arbitrary robot q on the SEC and moves cautiously toward it.

LEMMA 5.15. *Let the configuration at time t be still, one robot r be at c, $SA(\mathcal{R} \setminus \{c\})$ be simple, and $NI = 1$. Then there exists a time $t^* > t$ when the robots reach a still configuration $\mathcal{H}$ such that*

1. $\mathcal{H}$ *is regular biangular, or*
2. $\mathcal{H}$ *is dense, or*
3. $\mathcal{H}$ *is plain with no robot at c, $SA(\mathcal{R})$ is mixed, and only one robot is inside SEC.*

Proof. If r is the only robot inside SEC, by hypothesis it is at c. According to Routine4 only r is allowed to move: it cautiously moves toward a robot q that is on SEC. Note that as soon as r moves, the configuration becomes mixed; by Routine5 during this movement of r, all other robots stay still. The lemma follows the first time r stops.   □

If there are only two robots r and r' inside SEC (with r at c), then r' moves cautiously toward c.

LEMMA 5.16. *Let the configuration at time t be still, one robot r be at c, $SA(\mathcal{R} \setminus \{c\})$ be simple, and $NI = 2$. Then there exists a time $t^* > t$ when the robots reach a still configuration $\mathcal{H}$ that is either regular biangular or dense at c.*

Proof. If r and r' are the only robots inside SEC, according to Routine4 only r' is allowed to move: it cautiously moves toward r. Note that during the movements of r', no other robot is allowed to perform a nonnull movement. Let $t' > t$ be the first time r' stops after a nonnull movement. At time t', (i) r' stopped on a critical point where a regular biangular configuration is formed; or (ii) r' reached r at c; or (iii) r' stopped inside SEC on a point such that no biangular configuration is created. In cases (i) and (ii) the lemma holds. In case c, the conditions of the lemma are still met,

---

Routine 4 Subroutine for one robot r at c; $SA(\mathcal{R} \setminus \{c\})$ is simple.

$NI :=$ Number of robots inside SEC;

Case NI

- $NI = 1$ %r is the only robot inside SEC%
  $q :=$ Any arbitrary robot on SEC;
5:     **If** I Am at c **Then** moveCautiously(q).
- $NI = 2$
  $r' :=$ Robot inside SEC not at c;
  If I Am r' **Then** moveCautiously(c).
- $NI > 2$ %More than two robots are inside SEC%
10:     **If** $\mathcal{R}$ Is irregular periodic **Then**
        If I am inside SEC **Then** moveTo(c).
      Else
        $\overline{s} := |StS^+(\mathcal{R} \setminus \{c\}) \cup StS^-(\mathcal{R} \setminus \{c\})|$
        Case $\overline{s}$
15:         - $\overline{s} = 1$
            $l :=$ SelectOneInside($SA(\mathcal{R} \setminus \{c\})$);
            If I am l **Then** moveCautiously(c)
          - $\overline{s} = 2$
            $T :=$ SelectTeamInside($SA(\mathcal{R} \setminus \{c\})$);
20:           **If** $|T| = 1$ **Then**
              $l :=$ Robot in T;
              If I am l **Then** moveCautiously(c).
            Else %Two robots in T%
              $(l', l'') :=$ Robots in T;
25:             **If** I am l' or l'' **Then** movePairwiseCautiously(l', l'', c)
          - $\overline{s} > 2$
            If I am inside SEC **Then** moveTo(c).

---

with r' closer to r. Hence, by Assumption Dis, condition (i) or (ii) will eventually hold, and the lemma will follow. $\square$

Based on the previous lemma, we can now extend Lemma 5.7, showing that a still configuration is always reached.

LEMMA 5.17. *Let the configuration at time t be plain, no robot be at c, $SA(\mathcal{R})$ be simple, $s = 2$, only robots of one team T be inside SEC, with $T = \{l', l''\}$, and at least one robot of T be inside SEC. Furthermore, let the configuration be still, or* stillBut(l)*, $l \in T$, with l either acting on the median point on $Rad(l)$ and $|\mathcal{CP}((l', c), (l'', c))| = 1$ or performing* movePairwiseCautiously(l', l'', c)*. Then there exists a time $t^* > t$ when the robots reach a still configuration $\mathcal{H}$ such that either*

1. *$\mathcal{H}$ is biangular, or*
2. *$\mathcal{H}$ is dense at c.*

Proof. By Lemma 5.7, we already know that within finite time the robots reach a still configuration $\mathcal{H}'$ such that $\mathcal{H}'$ is regular biangular and still, or $\mathcal{H}'$ is dense at c and still, or in $\mathcal{H}'$ one robot of the team, say, l', is at c, $SA(\mathcal{R} \setminus \{c\})$ is simple, l'' is the only other robot inside SEC, and $\mathcal{H}'$ is plain and still or stillBut(l''), with l'' safely acting on c. In the first two cases the lemma holds. Thus, it suffices to prove that, under the conditions of the last case, the robots will reach a still configuration that is either regular biangular or dense at c.

According to `Routine4`, l'' is the only robot allowed to perform a nonnull move-ment. If $\mathcal{H}'$ is still, then by Lemma 5.16 and Property 7, the lemma follows. If $\mathcal{H}'$ is `stillBut`(l''), then l'' is acting on c. Let t' be the time when l''ends this move. At time t', either l'' reached l' at c, or l'' stopped inside SEC on a point such that no biangular configuration is formed. In the first case the lemma holds. In the latter case, the lemma follows by Lemma 5.16. $\quad\square$

Hence, the lemmas proven for case $s = 2$ (Lemmas 5.6–5.9 and 5.17) can be summarized by the following theorem.

THEOREM 5.18. *Let the configuration at time t be plain and still, no robot be at c, $SA(\mathcal{R})$ be simple, and $s = 2$. Then there exists a time $t^* > t$ when the robots reach a still configuration $\mathcal{H}$ such that*

    1. *$\mathcal{H}$ is biangular, or*
    2. *$\mathcal{H}$ is dense at c, or*
    3. *in $\mathcal{H}$ one robot is at c, $SA(\mathcal{R} \setminus \{c\})$ is simple, and no other robot is inside SEC.*

If more than two robots are inside SEC and $\mathcal{R}$ is irregular periodic (either this is an initial configuration, or this configuration has been created from the case $s > 2$ in section 5.3.3), all robots inside SEC move toward c.

LEMMA 5.19. *Let the configuration at time t be plain, $NI > 2$, with NI the number of robots inside SEC, with one robot r at c, $SA(\mathcal{R} \setminus \{c\})$ be simple, $\mathcal{R}$ be irregular periodic, and all robots inside SEC be from the same class T. Furthermore, let the configuration be still, or `stillBut`(T'), with $T' \subseteq T$, with all acting robots performing `moveTo`(c). Then there exists a time $t^* > t$ when the robots reach a configuration $\mathcal{H}$ such that $\mathcal{H}$ is dense at c and either still or `stillBut`(T''), $T'' \subseteq T$, where all acting robots are safely acting on c.*

Proof. By hypothesis and by `Routine4`, the robots that perform a nonnull move-ment can only act on c. Without loss of generality, let r' be the first robot to stop after a nonnull movement, say, at time $t' > t$. Since the movements of all robots can be only along their radii, during these movements the configuration remains plain and $SA(\mathcal{R} \setminus \{c\})$ does not change; hence, it is still *simple*. By Lemma 2.5, at time t', either (i) r' reaches r at c, or (ii) r' stops inside SEC at a point closer to c.

In all cases, at time t' all other robots are either still or safely acting on c. In case (i) the lemma holds. In case (ii), $SA(\mathcal{R} \setminus \{c\})$ is again simple and $\mathcal{R}$ is irregular periodic; hence, the hypotheses of the lemma are still met with r' at a closer distances to c. Hence, by Assumption DIS, condition (i) will eventually hold. Thus the lemma will eventually hold. $\quad\square$

Based on the previous lemma, we can now extend Lemma 5.12, showing that a dense configuration is always reached.

LEMMA 5.20. *Let the configuration at time t be plain, no robot be at c, at least two robots be inside SEC, all robots inside SEC be from the same class T, $SA(\mathcal{R})$ be simple, and $s > 2$. Furthermore, let the configuration be still, or `stillBut`(T'), with $T' \subseteq T$, where each acting robot is safely acting on either c or the median point of its radius. Then there exists a time $t^* > t$ when the robots reach a configuration $\mathcal{H}$ that is dense at c, and where all acting robots are safely acting on c.*

Proof. By Lemma 5.12, in finite time the robots reach a configuration $\mathcal{H}'$ such that either (i) $\mathcal{H}'$ is dense at c and all acting robots are safely acting on c, or (ii) $SA(\mathcal{R} \setminus \{c\})$ is simple and $\mathcal{R}$ is irregular periodic; inside SEC there are at least two robots, they are all from T, and one of them, say, r, is at c; and $\mathcal{H}'$ is plain, still or `stillBut`(T''), with $T'' \subseteq T \setminus \{r\}$, and with all robots in T'' safely acting on c. In

case (i) the lemma trivially holds. In case (ii), the lemma holds by Lemma 5.19. □

Hence, the lemmas proven for the case $s > 2$ (Lemmas 5.11–5.13 and 5.20) can be summarized by the following theorem.

THEOREM 5.21. *Let the configuration at time t be plain and still, no robot be at c, $SA(\mathcal{R})$ be simple, and $s > 2$. Then there exists a time $t^* > t$ when the robots reach a configuration $\mathcal{H}$ that is dense at c, with all acting robots safely acting on c.*

The last case considered by Routine4 is when more than two robots are inside SEC and $\mathcal{R}$ is *not* irregular periodic: in this case, the routine behaves similarly to Routine3, described in the previous section. The only difference is that in this case the robot at c does not move, and all the operations are done with respect to $SA(\mathcal{R} \setminus \{c\})$. In particular, Routine4 distinguishes three cases, depending on the value of $\overline{s}$, the cardinality of $StS^+(\mathcal{R} \setminus \{c\}) \cup StS^-(\mathcal{R} \setminus \{c\})$.

1. If $\overline{s} = 1$, then Routine4 elects a unique robot l that is inside SEC by calling SelectOneInside(). l moves cautiously toward c, while all the other robots do not move.
2. If $\overline{s} = 2$, then a team is elected by calling routine SelectTeamInside() (SelectTeamInside() here selects a team that has at least one robot inside SEC). If $|T| = 1$, then the only robot in T moves cautiously toward c, while the others wait; otherwise, the two robots in T move pairwise cautiously toward c, and the others stay still.
3. If $\overline{s} > 2$, then $SA(\mathcal{R} \setminus \{c\})$ is periodic (and all the considerations in section 5.3.3 apply also here). In this case, all robots inside SEC move toward c.

The following lemma handles the first case.

LEMMA 5.22. *Let at time t the robots be in a still configuration $\mathcal{H}$ such that $NI > 2$, with one robot r at c, $SA(\mathcal{R} \setminus \{c\})$ be simple and $\mathcal{R}$ not irregular periodic, and $\overline{s} = 1$. Then there exists a time $t^* > t$ when the robots reach a still configuration $\mathcal{H}'$ that is either biangular or dense.*

Proof. Since $\overline{s} = 1$, there is a unique direction for $LMS(\mathcal{R} \setminus \{c\})$, and it is possible to establish a total ordering of all robots in $\mathcal{R} \setminus \{r\}$. SelectOneInside() elects the first robot, say, l, that is inside SEC (by hypothesis, inside SEC there are at least two robots from $\mathcal{R} \setminus \{r\}$). According to Routine4, only l is allowed to move, while all other robots stay still during the cautious movement of l toward c. Let $t' > t$ be the first time when l stops after a nonnull movement. Since l moves radially, during its movements $SA(\mathcal{R} \setminus \{c\})$ stays simple and $\mathcal{R}$ not irregular periodic, and $\overline{s} = 1$. At time t', (i) l reaches r at c, or (ii) a biangular configuration is formed, or (iii) l stops on a point such that no biangular configuration is formed. In cases (i) and (ii) the lemma follows. In the last case (iii), the hypotheses of the lemma are still met, with l closer to c; hence, by Assumption DIS, condition (i) or (ii) will eventually hold, and the lemma will hold. □

Let us now consider the case when $\overline{s} = 2$.

LEMMA 5.23. *Let at time t the robots be in a still configuration such that $NI > 2$, with NI the number of robots inside SEC, with one robot r at c; and let $SA(\mathcal{R} \setminus \{c\})$ be simple and $\mathcal{R}$ not irregular periodic, with $\overline{s} = 2$. Then there exists a time $t^* > t$ when the robots reach a configuration $\mathcal{H}$ such that*

1. *$\mathcal{H}$ is biangular and still, or*
2. *$\mathcal{H}$ is dense at c and still, or*
3. *$\mathcal{H}$ is dense at c and stillBut(l'), where l' is a robot safely acting on c.*

Proof. The robots can be logically grouped into teams of at most two robots, and

it is possible to establish a total ordering of those teams (as described in section 5.3.2). In particular, `SelectTeamInside()` invoked within `Routine4` elects the first team T that has at least one robot inside SEC. We distinguish two cases, based on the cardinality of T.

1. If T is composed of just one robot, l, then, according to `Routine4`, l is the only robot allowed to move, while the others wait. Let $t' > t$ be the first time when l stops after a nonnull movement toward c. Note that, since l moves radially, during its movements $SA(\mathcal{R} \setminus \{c\})$ stays simple and $\mathcal{R}$ not irregular periodic, $\overline{s} = 2$, and $T = \{l\}$ is again the first team with at least one robot inside SEC. At time t', (i) l reaches r at c, or (ii) a biangular configuration is formed, or (iii) l stops on a point such that no biangular configuration is formed. In cases (i) and (ii) the lemma follows. In the last case (iii), the hypotheses of the lemma are still met, with l closer to c; hence, by Assumption DIS, condition (i) or (ii) will eventually hold, and the lemma will hold.

2. If $T = \{l', l''\}$ is composed of two robots, by `Routine4`, the only move operation they can perform is `movePairwiseCautiously`(l', l'', c). Without loss of generality, let l' be the first to stop after a nonnull movement, say, at time $t' > t$. During l'''s movement only l'' is allowed to move, by performing `movePairwiseCautiously`(l', l'', c). Since the movements of both robots can be only along $Rad(l')$ and $Rad(l'')$, respectively, during these movements the configuration remains plain and $SA(\mathcal{R} \setminus \{c\})$ does not change; hence, it is still *simple*, $\overline{s} = 2$, and the robots in T are the only ones allowed to move; this is also true if only l' moves during this time. By Properties 11 and 10, at time t', (i) l' and l'' are on a pair of critical points such that the configuration is biangular and they are both still, or (ii) l' and l'' are both on c and they are both still, or (iii) l' is on c, or (d) l' is at a point closer to c, and where no biangular configuration exists. In the last two cases, robot l'' is either still or performing `movePairwiseCautiously`(l', l'', c).

   In case (i), (ii), and (iii) the lemma holds. In case (d) the hypotheses of the lemma are still met and the distance of the team from c has decreased. Hence, by Assumption DIS, condition (i), (ii), or (iii) will eventually hold. Thus the lemma will eventually hold. $\square$

LEMMA 5.24. *Let the configuration at time t be plain, $NI > 2$, with NI the number of robots inside SEC, with one robot r at c, $SA(\mathcal{R} \setminus \{c\})$ be simple and $\mathcal{R}$ not irregular periodic, and $\overline{s} > 2$. Furthermore, let the configuration be still, or* `still-But`(T), *with $T \subseteq \mathcal{R}$ the robots inside SEC at time t, with all acting robots performing* `moveTo`(c). *Then there exists a time $t^* > t$ when the robots reach a configuration $\mathcal{H}$ such that $\mathcal{H}$ is dense at c and all acting robots are safely acting on c.*

Proof. In this case, according to `Routine4`, all robots that are inside SEC execute `moveTo(c)` on their radii. Note that, since $SA(\mathcal{R} \setminus \{c\})$ is simple and $\mathcal{R}$ is not irregular periodic at time t, no collision can occur during these movements. Moreover, by Lemma 2.5, no biangularity can occur. Without loss of generality, let r' be the first robot to stop after a nonnull movement, say, at time $t' > t$. Since the movements of all robots can be only along their radii, during these movements the configuration remains plain and $SA(\mathcal{R} \setminus \{c\})$ does not change; hence, it is still *simple*, $\overline{s} > 2$, and the robots in T are the only ones allowed to move. At time t', either (i) r' reaches r at c, or (ii) r' stops before reaching c, on a point closer to c.

In both cases, at time t' all other robots are either still or executing `moveTo(c)`.

In case (i) the lemma holds. In case (ii), the hypotheses of the lemma are still met and the distance of r' from c has decreased. Hence, by Assumption DIS, condition (i) will eventually hold. Thus the lemma will eventually hold. $\square$

Summarizing, by Lemmas 5.22–5.24, we have the following theorem.

THEOREM 5.25. *Let at time t the robots be in a plain and still configuration such that $NI > 2$, with one robot r at c, $SA(\mathcal{R} \setminus \{c\})$ be simple, and $\mathcal{R}$ not irregular periodic. Then there exists a time $t^* > t$ when the robots reach a configuration $\mathcal{H}$ such that either*

1. *$\mathcal{H}$ is biangular and still, or*
2. *$\mathcal{H}$ is dense at c and still, or* stillBut(T), *with $T \subset \mathcal{R} \setminus \{r\}$, and all acting robots are safely acting on c.*

5.5. No robot at the center of SEC and $SA(\mathcal{R})$ is mixed. Let us consider now the case when *no robot is at the center* of SEC, and $SA(\mathcal{R})$ is mixed. Recall that if $SA(\mathcal{R})$ is mixed, then there is at least one radius of SEC where more than one robot lies. Therefore, by definition of $SA(\mathcal{R})$, the lexicographically minimal string of angles always starts with an angle whose width is zero. Moreover, on each radius with at least two robots, at least one is already inside SEC.

Routine 5 Subroutine: No robot at c, SA mixed.

Case s
- $s = 1$
  $l :=$ SelectOneInside();
  $q :=$ robot on the radius of l closest to l;
  5:   **If** I am l **Then** moveCautiously(q).
- $s = 2$
  $(l', l'') :=$ SelectTeamInside();
  If I am l' or l'' **Then** movePairwiseCautiously(l', l'', c).
- $s > 2$
  10:   $T :=$ SelectClassInside();
  If I am in T **Then** moveTo(c)

As usual, we consider different cases depending on the value of s. Let $s = 1$ and $SA(\mathcal{R})$ be mixed (note that this situation could be originated from the one described in Lemma 5.15). In this case we elect the first robot l in the total order implied by $LMS(\mathcal{R})$: l moves cautiously toward q, the robot closest to l that lies on $Rad(l)$.

LEMMA 5.26. *Let at time t the robots be in a still configuration such that no robot is at c, $SA(\mathcal{R})$ is mixed, and $s = 1$. Then there exists a time $t^* > t$ when the robots reach a still configuration $\mathcal{H}$ that is either biangular or dense.*

Proof. By hypotheses, it is possible to establish a total ordering of the robots; SelectOneInside() returns the first robot l in this ordering. Since $SA(\mathcal{R})$ is mixed, at least one other robot lies on $Rad(l)$, and l is the robot on $Rad(l)$ closest to c; let q be the robot closest to l on $Rad(l)$. According to Routine5, l is the only one allowed to move: it moves cautiously toward q. Let t' be the first time when l stops after a nonnull move; during l's movements, no other robot can move. Moreover, since l moves radially, $SA(\mathcal{R})$ stays mixed, with $s = 1$, and l remains the first robot in the total order implied by $LMS(\mathcal{R})$. Thus, at time t', (i) l reaches q, or (ii) a biangular configuration is created, or (iii) l stops on a point on $Rad(l)$ where no biangularity occurs. In cases (i) and (ii) the lemma follows. In case (iii), the hypotheses of the

lemma are still met, with l closer to q; therefore, by Assumption DIS, condition (i) or (ii) will eventually hold, and the lemma follows. □

Let us now consider the case when $s = 2$. The robots can be grouped in teams of at most two robots, and it is possible to establish a total ordering on the teams, as described in section 5.3.2.

LEMMA 5.27. *Let the configuration at time t be plain, no robot be at c, $SA(\mathcal{R})$ be mixed, $s = 2$, and T be the first team in the total order implied by $SA(\mathcal{R})$. Then, $T = \{l', l''\}$ is composed of two distinct robots, both inside SEC. Furthermore, let the configuration be still, or* stillBut(l)*, $l \in T$, with l performing* movePairwise-Cautiously(l', l'', c)*. Then there exists a time $t^* > t$ when the robots reach a configuration $\mathcal{H}$ such that*

1. *$\mathcal{H}$ is biangular and still, or*
2. *$\mathcal{H}$ is dense at c and still, or*
3. *in $\mathcal{H}$ one robot of T, say, l', is at c; $SA(\mathcal{R})$ is mixed; and $\mathcal{H}$ is plain and still or* stillBut(l'')*, with l'' performing* movePairwiseCautiously(l', l'', c)*.*

Proof. Let T be the team selected by SelectTeamInside(). By Lemma 2.1, by definition of $LMS(\mathcal{R})$, and by the ranking defined in section 5.3.2 used to form the teams of robots, it follows that the first team in the total ordering is composed of two distinct robots, l' and l'', both inside *SEC*. Moreover, since $SA(\mathcal{R})$ is mixed, they are both inside *SEC*, and the closest robots to c on $Rad(l')$ and $Rad(l'')$, respectively.

By Routine5 and by hypothesis, the only move operation l' and l'' can perform is movePairwiseCautiously(l', l'', c), while all the others are waiting. Without loss of generality, let l' be the first to stop after a nonnull movement, say, at time $t' > t$. Since the movements of both robots can only be along $Rad(l')$ and $Rad(l'')$, respectively, and there are no robots on $\overline{l'c}$ and $\overline{l''c}$, any collision (hence creation of unintended dense points) is avoided; hence, during these movements the configuration remains plain and $SA(\mathcal{R})$ does not change. Therefore, it is still mixed, $s = 2$, and the robots in T are the only ones allowed to move. By Property 11, at time t', (i) l' and l'' are on a pair of critical points such that the configuration is biangular, and they are both still, or (ii) l' and l'' are both still and on c, or (iii) l' is on c, or (iv) no robot is on c, l' is at a point closer to c and where no biangular configuration exists. In the last two cases, at time t', robot l'' is either still or performing movePairwiseCautiously(l', l'', c).

In cases (i) and (ii) the lemma holds. In case (iii), $SA(\mathcal{R} \setminus \{c\})$ is mixed. Moreover, l'' is either still or moving toward c executing operation movePairwiseCautiously(l', l'', c): in both cases the lemma holds. In case (iv) the hypotheses of the lemma are still met and the distance of the team from c has decreased. Hence, by Assumption DIS, condition (i), (ii), or (iii) will eventually hold. Thus the lemma will eventually hold. □

Finally, let us consider the case when $s > 2$. The robots can be grouped in equivalence classes, and it is possible to establish a total ordering on these classes, as described in section 5.3.3.

LEMMA 5.28. *Let the configuration at time t be plain, no robot be at c, $SA(\mathcal{R})$ be mixed, $s > 2$, and T be the first class of equivalence in the order implied by $LMS(\mathcal{R})$. Then all robots of T are inside SEC. Furthermore, let the configuration be still, or* stillBut(T')*, with $T' \subseteq T$, with all acting robots safely acting on c. Then there exists a time $t^* > t$ when the robots reach a configuration $\mathcal{H}$ such that either*

1. *$\mathcal{H}$ is dense at c and all acting robots are safely acting on c, or*
2. *$SA(\mathcal{R} \setminus \{c\})$ is mixed and $\mathcal{R}$ is irregular periodic; one of the robots from T, say, r, is at c; and $\mathcal{H}$ is plain, still or* stillBut(T'')*, with $T'' \subseteq T \setminus \{r\}$,*

and all acting robots are safely acting on c.

Proof. Let T be the class selected by `SelectClassInside()` at time t. Since $SA(\mathcal{R})$ is mixed, by Lemma 2.1, by definition of $LMS(\mathcal{R})$, and by the equivalence classes defined in section 5.3.3, it follows that all the robots of the first class are inside SEC. Moreover, since $SA(\mathcal{R})$ is mixed, all of them are the closest robots to c on their respective radii.

By `Routine5` and by hypothesis, the only move operation the robots in T can perform is `moveTo(c)`. Without loss of generality, let r be the first robot to stop after a nonnull movement, say, at time $t' > t$. During r's movement only robots in T are allowed to move, by performing `moveTo(c)`. Since the movements of all robots can be only along their radii, during these movements any collision is avoided since there are no robots on $\overline{r_i c}$ for all $r_i \in T$; thus the configuration remains plain and $SA(\mathcal{R})$ does not change; hence, it is still *simple*, $s > 2$, and the robots in T are the only ones allowed to move. At time t', (i) two robots from T reach simultaneously c and they are both still or (ii) one robot in T reaches c, or (iii) no robot is on c, and r is at a point closer to c.

In all cases, at time t' all other robots in T are either still or safely acting on c. In case (i) the lemma holds. In case (ii), $SA(\mathcal{R} \setminus \{c\})$ is mixed and $\mathcal{R}$ is irregular periodic; hence, the lemma follows. In case (iii) the hypotheses of the lemma are still met and the distance of r from c has decreased. Hence, by Assumption DIS, condition (i), (ii), or (iii) will eventually hold. Thus the lemma will eventually hold. ☐

5.6. A robot at the center of SEC and $SA(\mathcal{R} \setminus \{c\})$ is mixed. First notice that when $SA(\mathcal{R} \setminus \{c\})$ is mixed, in addition to the robot r at the center, there is at least one other robot r' inside SEC.

If $\mathcal{R}$ is irregular periodic, `Routine6` elects the first class in this order; only these robots are allowed to move toward c.

Routine 6 Subroutine for one robot r at c, $SA(\mathcal{R} \setminus \{c\})$ mixed.

   If $\mathcal{R}$ Is irregular periodic **Then**
      $T :=$ `SelectClassInside()`;
      If $|T| = 2$ and I am in T **Then** `moveCautiously(c)`.
      If $|T| > 2$ and I am in T **Then** `moveTo(c)`.
5: **Else**
      $\overline{s} := |StS^+(\mathcal{R} \setminus \{c\}) \cup StS^-(\mathcal{R} \setminus \{c\})|$
      Case $\overline{s}$
         • $\overline{s} = 1$
           $l :=$ `SelectOneInside()`;
10:          **If** I am l **Then** `moveCautiously(c)`.
         • $\overline{s} = 2$
           $(l_1, l_2) :=$ `SelectTeamInside()`;
           If I am l_1 or l_2 **Then** `movePairwiseCautiously(`l_1, l_2, c`)`.
         • $\overline{s} > 2$
15:          **If** More than one robot on my radius and I am the closest to c **Then** `moveTo(c)`

LEMMA 5.29. *Let the configuration at time t be plain, with one robot r at c, $SA(\mathcal{R} \setminus \{c\})$ be mixed, and $\mathcal{R}$ be irregular periodic. Furthermore, let the configuration be still, or* `stillBut(T)`, *with T the smallest class inside SEC and with all acting robots safely acting on c. Then there exists a time $t^* > t$ when the robots reach a*

configuration $\mathcal{H}$ such that $\mathcal{H}$ is dense at c and either still or stillBut(T'), $T' \subseteq T$, *where all acting robots are safely acting on c.*

Proof. Following the considerations made in section 5.3.3, the robots can be grouped into equivalence classes, and it is possible to establish a total ordering on the classes. By Lemma 2.1, by definition of $LMS(\mathcal{R} \setminus \{c\})$, and by the equivalence classes defined in section 5.3.3, it follows that all the robots of the first class are inside SEC. Moreover, each of them is the closest robot to c on its radius. Note also that having $|T| = 1$ implies that there is no equivalent robot for the only robot in T; that is, $\mathcal{R}$ cannot be periodic and hence cannot be irregular periodic. Routine6 distinguishes two cases.

1. If $|T| = 2$, then in T there are two robots: one of them is the one at c, while the other, say, r, is allowed to cautiously move toward c. Let t' be the first time when r stops after a nonnull movement. During r's movement no other robot is allowed to move. Since $SA(\mathcal{R} \setminus \{c\})$ is mixed, there are no robots on $\overline{rc}$; hence, during these movements any collision is avoided; thus the configuration remains plain and $SA(\mathcal{R} \setminus \{c\})$ does not change; hence, it is still mixed, and $\mathcal{R}$ is irregular periodic, with r the only robot allowed to move. At time t', (i) r reaches c, or (ii) r stops on a point such that the configuration becomes biangular, or (iii) only one robot is on c, and r is at a point closer to c where no biangular configuration is reached.

2. If $|T| > 2$, by Routine6 and by hypothesis, the only move operation the robots in T can perform is moveTo(c); by Lemma 2.5, the configuration cannot become biangular during these movements. Without loss of generality, let r be the first robot to stop after a nonnull movement, say, at time $t' > t$. During r's movement only robots in T are allowed to move, by performing moveTo(c). Since $SA(\mathcal{R} \setminus \{c\})$ is mixed, there are no robots on $\overline{r_i c}$ for all $r_i \in T$; furthermore, the movements of all robots can be only along their radii; hence, during these movements any collision is avoided; thus the configuration remains plain and $SA(\mathcal{R} \setminus \{c\})$ does not change; hence, it is still mixed, $\mathcal{R}$ is irregular periodic, and the robots in T are the only ones allowed to move. At time t', either (iv) one robot in T reaches c, or (v) only one robot is on c, and r is at a point closer to c.

In all cases, at time t' all other robots in T are either still or safely acting on c. In cases (i), (ii), and (iv) the lemma holds. In cases (iii) and (v), the hypotheses of the lemma are still met and the distance of r from c has decreased. Hence, by Assumption DIS, condition (i), (ii), or (iv) will eventually hold. Thus the lemma will eventually hold. $\quad\square$

Furthermore, we can now extend Lemma 5.27, proving that a still configuration is always reached.

LEMMA 5.30. *Let the configuration at time t be plain, no robot be at c, $SA(\mathcal{R})$ be mixed, $s = 2$, and T be the first pair in the total order implied by $SA(\mathcal{R})$. Then, $T = \{l', l''\}$ is composed of two distinct robots, both inside SEC. Furthermore, let the configuration be still, or* stillBut(l), $l \in T$, *with l performing* movePairwise- Cautiously(l', l'', c). *Then there exists a time $t^* > t$ when the robots reach a still configuration $\mathcal{H}$ such that either*

1. $\mathcal{H}$ *is biangular, or*
2. $\mathcal{H}$ *is dense at c.*

Proof. By Lemma 5.27, the only case that has to be considered is when only one among l' and l'' reaches c, say, l', while the other is acting on c. Let t' be

the first time when l' reaches c. Note that, at time t, $\mathcal{R}$ is periodic; at time t', $\mathcal{R}$ becomes irregular periodic, with the class returned by `SelectClassInside()` having two robots, and $SA(\mathcal{R} \setminus \{c\})$ mixed. Hence, by `Routine6`, l'' is the only robot allowed to move at time t': it cautiously moves toward c. Let t'' be the first time l'' stops after a nonnull movement. We have three possible cases at time t'': (i) l'' is not at c at a position forming a biangular configuration; (ii) l'' is at c; (iii) l'' is inside SEC not at c and the configuration is not biangular. In cases (i) and (ii) the lemma immediately follows. In case (iii), according to `Routine6`, only l'' is allowed to move: it can only cautiously move toward c. Moreover, all other robots do not move during the cautious movements of l''. The lemma follows by previous Lemma 5.29. $\square$

Similarly, we extend Lemma 5.28.

LEMMA 5.31. *Let the configuration at time t be plain, no robot be at c, $SA(\mathcal{R})$ be mixed, $s > 2$, and T be the first class of equivalence in the order implied by $LMS(\mathcal{R})$. Then all robots of T are inside SEC. Furthermore, let the configuration be still, or* `stillBut(T')`*, with $T' \subseteq T$, with all acting robots performing* `moveTo(c)`*. Then there exists a time $t^* > t$ when the robots reach a configuration $\mathcal{H}$ that is dense at c and still, or* `stillBut(T'')`*, with $T'' \subseteq T \setminus \{r\}$ and $r \in T$ the robot at c, and with all acting robots safely acting on c.*

Proof. By Lemma 5.28, the only case we have to analyze is when, at time $t' > t$, only one robot r from T reaches c, while all the others in T are safely acting on c. At time t', $SA(\mathcal{R} \setminus \{c\})$ is clearly mixed, and $\mathcal{R}$ is irregular periodic with $|T| > 2$. Therefore, the lemma follows by Lemma 5.29. $\square$

The last case we need to consider is when $\mathcal{R}$ is not irregular periodic; in this case, `Routine6` distinguishes several cases according to the value of $\overline{s}$.

If $\overline{s} = 1$, the first robot in the total ordering implied by $LMS(\mathcal{R} \setminus \{c\})$ is elected as a leader and moves cautiously toward the center.

LEMMA 5.32. *Let at time t the robots be in a still configuration $\mathcal{H}$ such that one robot r is at c, $SA(\mathcal{R} \setminus \{c\})$ be mixed and $\mathcal{R}$ not irregular periodic, and $\overline{s} = 1$. Then there exists a time $t^* > t$ when the robots reach a still configuration $\mathcal{H}'$ that is either biangular or dense at c.*

Proof. Following the considerations in section 5.3.1, in this case there is a unique direction for $LMS(\mathcal{R} \setminus \{c\})$, and it is possible to establish a total ordering of all robots in $\mathcal{R} \setminus \{r\}$. `SelectOneInside()` elects the first robot, say, l: since $SA(\mathcal{R} \setminus \{c\})$ is mixed, l is inside *SEC*. This case is similar to the case $\overline{s} = 1$ of `Routine4`, with the only difference that here $SA(\mathcal{R} \setminus \{c\})$ is mixed; however, by Lemma 2.1 and by definition of $LMS(\mathcal{R})$, l is the closest robot to c on its radius; hence no collisions (hence, unintended dense points) can be created during its movement toward c. The lemma follows similarly to Lemma 5.22. $\square$

Let us now consider the case when $\overline{s} = 2$. Using the team ordering introduced in section 5.3.2, `Routine6` elects a leader team; the robots in this team move pairwise-cautiously toward c.

LEMMA 5.33. *Let the configuration at time t be plain, one robot be at c, $SA(\mathcal{R} \setminus \{c\})$ be mixed and $\mathcal{R}$ not irregular periodic, $\overline{s} = 2$, and T be the first pair in the total order implied by $SA(\mathcal{R} \setminus \{c\})$. Then, $T = (l', l'')$ is composed of two distinct robots. Furthermore, let the configuration be still, or* `stillBut(l)`*, $l \in T$, with l performing* `movePairwiseCautiously(l', l'', c)`*. Then there exists a time $t' > t$ when the robots reach a configuration $\mathcal{H}$ such that*

1. *$\mathcal{H}$ is biangular and still, or*
2. *$\mathcal{H}$ is dense at c and still, or*

3. $\mathcal{H}$ *is dense at c with one robot of the team, say, l', at c; and $\mathcal{H}$ is* `still-But`(l'')*, with l'' performing* `movePairwiseCautiously`(l', l'', c).

Proof. Let T be the team selected by `SelectTeamInside()`. By Lemma 2.1, by definition of $LMS(\mathcal{R} \setminus \{c\})$, and by the ranking defined in section 5.3.2 used to form the teams of robots, it follows that the first team in the total ordering is composed of two distinct robots, l' and l''. Moreover, they are the closest robots to c on $Rad(l')$ and $Rad(l'')$, respectively. Note that this case is similar to the case handled in `Routine5`, when $s = 2$, $|T| = 2$. The only difference is that here there is already a robot at c. The proof follows the same reasoning as that of Lemma 5.27. □

Finally, if $\overline{s} > 2$, from each radius with more then one robot on it, the robot closest to c is chosen to move toward c. By Lemma 2.5, no biangularity can occur during these movements.

LEMMA 5.34. *Let the configuration at time t be plain, one robot r be at c, $SA(\mathcal{R} \setminus \{c\})$ be mixed and $\mathcal{R}$ not irregular periodic, and $\overline{s} > 2$. Furthermore, let the configuration be still, or* `stillBut`(T)*, with $T \subseteq \mathcal{R}$, with all acting robots performing* `moveTo`(c)*. Then there exists a time $t' > t$ when the robots reach a configuration $\mathcal{H}$ such that $\mathcal{H}$ is dense at c and all acting robots are safely acting on c.*

Proof. In this case, all robots that are inside *SEC*, and that are the closest to c on their radius, execute `moveTo(c)`. Note that this case is similar to the one handled by `Routine4` when $\overline{s} > 2$. The only difference is that here $SA(\mathcal{R} \setminus \{c\})$ is mixed. However, since `Routine6` allows to move only the robots that are inside *SEC* and that are the closest to c on their radius, no collision can occur. Moreover, by Lemma 2.5, no biangularity can occur. The proof follows similarly to the proof of Lemma 5.24. □

5.7. Correctness of Algorithm GOGATHER. Finally, in this section we prove the overall correctness of Algorithm GOGATHER (refer also to Figure 5.3).

THEOREM 5.35. *From any initial configuration $\mathcal{H}$, within a finite number of cycles, the robots reach a configuration that is biangular (regular or irregular) and still, or dense and still, or dense with all acting robots safely acting on the dense point.*

Proof. Consider the partition of the set of all possible initial configurations into classes, as shown in Figure 5.4. The proof will consider different cases, depending on which class the initial configuration $\mathcal{H}$ belongs to. First observe that, since $\mathcal{H}$ is initial, it is still and plain by definition.

Case 1. [*Biangular (regular or irregular)*] The theorem trivially holds.

Case 2.I.A.i. [*No robot is at c, $SA(\mathcal{R})$ is simple, and $s = 1$*] By Theorem 5.5, Lemma 5.15, and Lemma 5.26, in a finite number of cycles the robots reach a still configuration that is biangular or dense, and the theorem follows.

Case 2.I.A.ii. [*No robot is at c, $SA(\mathcal{R})$ is simple, and $s = 2$*] By Theorem 5.18, Lemma 5.15, and Lemma 5.26, in a finite number of cycles the robots reach a configuration that is either still and biangular or still and dense, and the theorem follows.

Case 2.I.A.iii. [*No robot is at c, $SA(\mathcal{R})$ is simple, and $s > 2$*] By Theorem 5.21, in a finite number of cycles the robots reach a configuration that is dense with all acting robots safely acting on c, and the theorem follows.

Case 2.II.A.i. [*One robot is at c, $SA(\mathcal{R} \setminus \{c\})$ is simple, and $NI \leq 2$, with NI the number of robots inside SEC*] By Lemmas 5.15 and 5.26 (when $NI = 1$), and by Lemma 5.16 (when $NI = 2$), in a finite number of cycles the robots reach a still configuration that is either biangular or dense, hence the theorem follows.
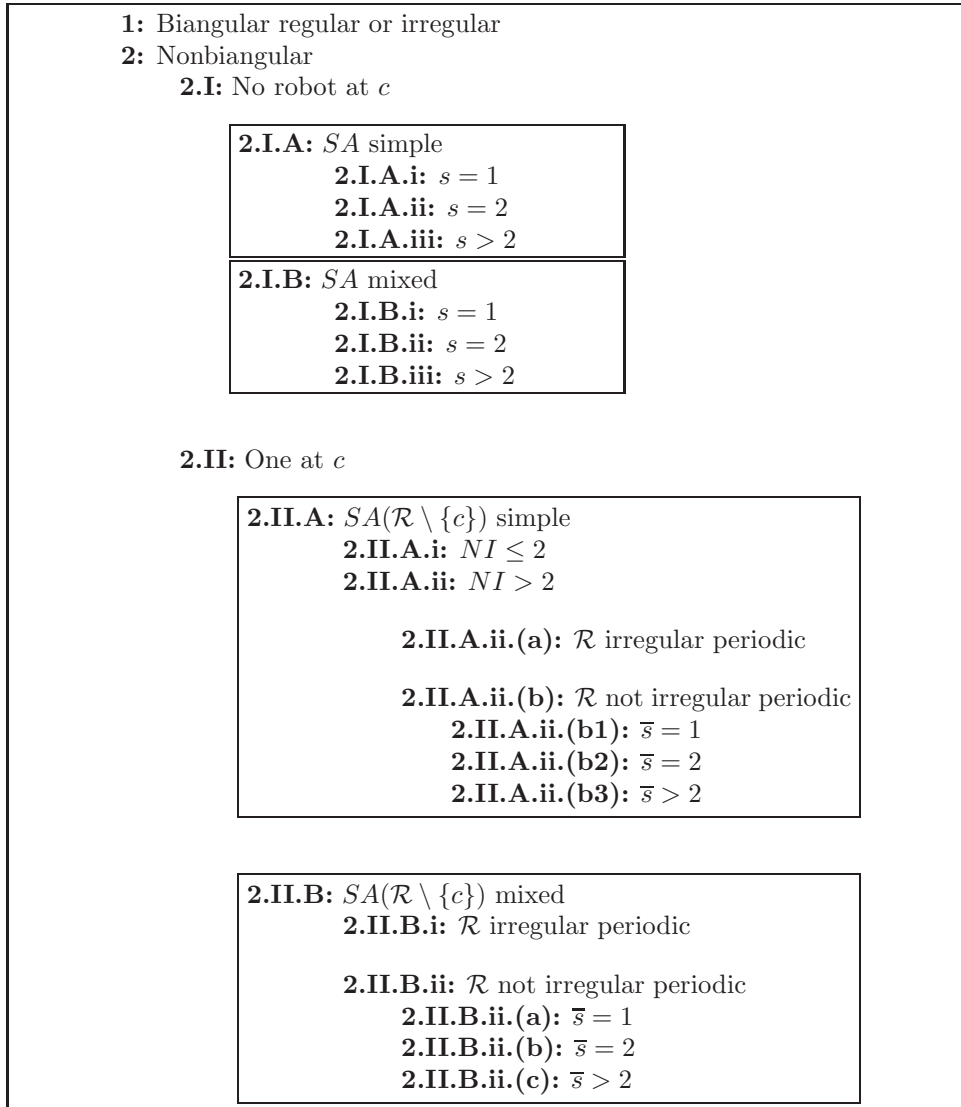
1: Biangular regular or irregular
2: Nonbiangular
    2.I: No robot at c

        2.I.A: SA simple
            2.I.A.i: $s = 1$
            2.I.A.ii: $s = 2$
            2.I.A.iii: $s > 2$

        2.I.B: SA mixed
            2.I.B.i: $s = 1$
            2.I.B.ii: $s = 2$
            2.I.B.iii: $s > 2$

    2.II: One at c

        2.II.A: $SA(\mathcal{R} \setminus \{c\})$ simple
            2.II.A.i: $NI \leq 2$
            2.II.A.ii: $NI > 2$

                2.II.A.ii.(a): $\mathcal{R}$ irregular periodic

                2.II.A.ii.(b): $\mathcal{R}$ not irregular periodic
                    2.II.A.ii.(b1): $\overline{s} = 1$
                    2.II.A.ii.(b2): $\overline{s} = 2$
                    2.II.A.ii.(b3): $\overline{s} > 2$

        2.II.B: $SA(\mathcal{R} \setminus \{c\})$ mixed
            2.II.B.i: $\mathcal{R}$ irregular periodic

            2.II.B.ii: $\mathcal{R}$ not irregular periodic
                2.II.B.ii.(a): $\overline{s} = 1$
                2.II.B.ii.(b): $\overline{s} = 2$
                2.II.B.ii.(c): $\overline{s} > 2$

FIG. 5.4. *All possible initial configurations.*

Case 2.II.A.ii.(a). [*One robots is at c, $SA(\mathcal{R} \setminus \{c\})$ is simple, $NI > 2$, and $\mathcal{R}$ is irregular periodic*] By Lemma 5.19, in a finite number of cycles the robots reach a configuration that is dense at c, with all acting robots safely acting on c.

Case 2.II.A.ii.(b). [*One robots is at c, $SA(\mathcal{R} \setminus \{c\})$ is simple, $NI > 2$, and $\mathcal{R}$ is not irregular periodic*] By Theorem 5.25, in a finite number of cycles the robots reach a configuration that is either still and biangular or dense at c with all acting robots safely acting on c.

Cases 2.I.B.i *and* 2.I.B.ii. [*No robot is at c, $SA(\mathcal{R})$ is mixed, with $s \leq 2$*] By Lemma 5.26 (when $s = 1$), and by Lemma 5.30 (when $s = 2$), in a finite number of cycles the robots reach a still configuration that is either biangular dense at

c, and the theorem follows.

Case 2.I.B.iii. [*No robot is at c, $SA(\mathcal{R})$ is mixed, and $s > 2$*] By Lemma 5.31, in a finite number of cycles the robots reach a configuration that is dense at c, with all acting robots (if any) safely acting on c, and the lemma follows.

Case 2.II.B.i. [*One robot is at c, $SA(\mathcal{R} \setminus \{c\})$ is mixed, and $\mathcal{R}$ is irregular periodic*] By Lemma 5.29 in a finite number of cycles the robots reach a configuration that dense at c, with all acting robots (if any) safely acting on c.

Case 2.II.B.ii.(a). [*One robot at c, $SA(\mathcal{R} \setminus \{c\})$ is mixed, $\mathcal{R}$ is not irregular periodic, and $\overline{s} = 1$*] By Lemma 5.32, in a finite number of cycles the robots reach a still configuration that is either biangular or dense at c.

Case 2.II.B.ii.(b). [*One robot is at c, $SA(\mathcal{R}\setminus\{c\})$ is mixed, $\mathcal{R}$ is not irregular periodic, and $\overline{s} = 2$*] By Lemma 5.33, in a finite number of cycles the robots reach a configuration that is still and biangular, or dense at c with all acting robots (if any) safely acting on c.

Case 2.II.B.ii.(c). [*One robot is at c, $SA(\mathcal{R}\setminus\{c\})$ is mixed, $\mathcal{R}$ is not irregular periodic, and $\overline{s} > 2$*] By Lemma 5.34, in a finite number of cycles the robots reach a configuration that is dense at c with all acting robots (if any) safely acting on c. □

By Assumption DIS and by Theorems 5.1, 5.2, and 5.35, the next theorem follows.

THEOREM 5.36. *Algorithm* GOGATHER *allows $n \geq 5$ asynchronous oblivious disoriented robots to gather within finite time starting from any initial plain configuration.*

6. Open problems. We have considered robots that are anonymous, asynchronous, oblivious, silent, and disoriented; and we have presented the first deterministic algorithm for the GATHERING PROBLEM of such robots that works for any initial configuration. The outstanding research question is now whether it is possible to solve gathering with even weaker robots.

As in [8, 27, 52, 56], we have assumed that each robot is within the visibility range of the others. The setting when this is not necessarily the case is referred to as *limited visibility*. Gathering protocols do exist with limited visibility, but only in models stronger than the one studied in this paper [3, 26, 29, 41, 45, 44, 49, 57]. The question of whether gathering can be solved when the robots with limited visibility are also oblivious, disoriented, and asynchronous, as considered here, is still unanswered.

The model can be weakened also by considering the possibility of *faults* and of *inaccuracies*. Solutions exist considering robots' failures and when movements and sensors' measurements are inaccurate; however, they assume stronger models (e.g., semisynchronous) [1, 12, 55, 30] or a simpler problem (CONVERGE) [12] or restricted environments (the one-dimensional space) [12]. In particular, no studies exist when the computations are subject to imprecisions. An important open research direction is to consider perturbations in the operating behavior of the robots or their sensors and to design fault-tolerant solutions to the GATHERING PROBLEM for the model considered here.

In all existing investigations on the GATHERING PROBLEM, the focus has been on computability (i.e., feasibility), while the *complexity* of the solutions has never been an issue; indeed, there is a general absence of cost measures. An interesting fundamental research question is the definition of cost measures and their use in the analysis of the complexity of the solution protocols. A step in this direction has been taken recently, analyzing the number of rounds for the (simpler) CONVERGE problem in the case of synchronous robots [19].

Appendix A. Proofs of basic properties.

Proof of Lemma 2.1. Without loss of generality, let us assume that $LMS(P, c) = \langle \alpha_0, \ldots, \alpha_{n-1} \rangle$. By contradiction, let us assume that $SA^+(P, c)[k] = SA^-(P, c)[k] = LMS(P, c)$. Then $\langle \alpha_0, \ldots, \alpha_{n-1} \rangle = \langle \alpha_{n-1}, \ldots, \alpha_0 \rangle$; that is, $LMS(P, c)$ is a palindrome. Let j be the first index such that $\alpha_j > \alpha_0$ (such an index exists by hypothesis); that is, $\alpha_{n-i-1} = \alpha_i = \alpha_0, 0 \leq i \leq j - 1$. It follows that $\langle \alpha_{n-j}, \alpha_{n-j+1}, \ldots, \alpha_{n-1}, \alpha_0, \alpha_1, \ldots, \alpha_{n-j-1} \rangle \langle_{lex} \langle \alpha_0, \alpha_1, \ldots, \alpha_{n-1} \rangle = LMS(P, c)$, which is a contradiction. □

Proof of Lemma 2.2. Let us consider the first team defined by the total order (refer to Figure 2.3). Notice that this team consists of two distinct robots, r_k and r_w. Consider also the smallest arc of circumference between them (call it arc); let arc' be the greatest arc of circumference between them. If the angle between r_k and r_w with respect to c is $180°$, then clearly the removal of any other team will leave SEC unchanged; since $n \geq 5$, there exists at least one other team composed of two distinct robots, hence the lemma follows.

Otherwise, since $n \geq 5$, there is at least a team on arc'. If there is also a team on arc, then the removal of r_k and r_w will leave SEC unchanged, and the lemma follows (refer to Figure 2.3(b)).

If no robot is on arc, then, since $n \geq 5$, there are at least two teams on arc'. The first of these two teams (i.e., the second team in the total order) must be composed of two distinct robots, r_1 and r_2 (refer to Figure 2.3(c)); while the second one (the first team on the arc of circumference defined by r_1 and r_2 that does not contain r_k and r_w) can be composed of either one single robot or of two distinct robots. In any case, the removal of r_1 and r_2 leaves SEC unchanged, and the lemma follows. □

Proof of Lemma 2.3. Consider two successive starting points x, y of $LMS(P, c)$ in $SA(P, c)$, where $x = p_i$ is the starting point of $SA^+(P, c)[i]$ and $y = p_j$ is the starting point of $SA^+(P, c)[j]$; and let $W = \langle \alpha_i, \alpha_{i+1}, \ldots, \alpha_{j-1} \rangle$; in other words, the first $|W|$ elements of $LMS(P, c)$ are precisely W. Since x is chosen arbitrarily, it follows that $LMS(P, c) = W^k$. Hence (1) holds, and (2) immediately follows. □

Proof of Lemma 2.4. If all points are on $SEC(P)$, then for any robot p, by Property 3, the points in $EQ(p)$ form a $\frac{n}{k}$-gon with all vertices on SEC. Thus, the lemma follows. □

Proof of Lemma 2.5. Let P_{SEC} be the set of points obtained by moving all points in P on $SEC(P)$. Since P is periodic, P_{SEC} is rotational symmetric with symmetry center c. Hence, by Property 4, c is the Weber point of P_{SEC}. Let P' be now the set of points obtained from P_{SEC} by moving any of the points in P_{SEC} towards c; again, by Property 2, c is the Weber point of P'. The claim follows since the Weber point is unique and by Property 4: moving some of the point toward it cannot make the set of points biangular, if it was not so before. □

Proof of Lemma 2.6. Let us assume that $P = p_0, \ldots, p_{n-1}$ is regular biangular. Then, by Property 4, its center of biangularity b is also the Weber point of P. Moreover, by Property 2, b stays the center of biangularity of P even if any point $p \in P$ is moved on p', with $p' \in \overline{p, b}$. In particular, b does not change if p is moved on b. In this case, P becomes irregular biangular with center b. The unicity of b follows from the unicity of the Weber point. □

Proof of Property 5. If P is biangular with one gap, then by definition there exist α and β such that $SA(P, b) = \langle (\alpha, \beta)^{\frac{n}{2}-1}, \gamma \rangle$, where $\gamma = \alpha + \beta$. This means that there exists a subset $S \subset P$ of the points, with $|S| = n/2$, such that S is equiangular, and $SA^+(S, b) = \langle \gamma^{\frac{n}{2}} \rangle$. □

Proof of Property 6. Let P be biangular with two gaps. Then, by definition, there

exist angles α and β such that

1. $SA^+(P, b) = W_1 = \langle \theta, (\beta, \alpha)^{\frac{n}{2}-2}, \beta \rangle$ where $\theta = 2\alpha + \beta$; or
2. $SA^+(P, b) = W_2 = \langle \gamma, (\alpha, \beta)^{k_1}, \gamma, (\alpha, \beta)^{k_2} \rangle$ with $k_1 + k_2 = n/2 - 2$ where $\gamma = \alpha + \beta$; or
3. $SA^+(P, b) = W_3 = \langle \gamma, (\alpha, \beta)^{k_1}, \alpha, \gamma, (\beta, \alpha)^{k_2}, \beta \rangle$, with $k_1 + k_2 = n/2 - 3$ where $\gamma = \alpha + \beta$.

Let us consider the three possible cases separately.

1. $W_1 = \langle \theta, (\beta, \alpha)^{\frac{n}{2}-2}, \beta \rangle = \langle \beta, (\alpha+\beta+\alpha), (\beta, \alpha)^{\frac{n}{2}-2} \rangle = \langle 2\gamma, \gamma^{\frac{n}{2}-2} \rangle = \langle \delta, \gamma^{\frac{n}{2}-2} \rangle$ where $\gamma = \alpha + \beta$ and $\delta = 2\gamma$. This means that there exists a subset $S \subset P$ of the points, with $|S| = n/2 - 1$, such that S is equiangular with one gap, and $SA^+(S, b) = \langle \delta, \gamma^{\frac{n}{2}-2} \rangle$.
2. $W_2 = \langle \gamma, (\alpha, \beta)^{k_1}, \gamma, (\alpha, \beta)^{k_2} \rangle = \gamma^{\frac{n}{2}}$. This means that there exists a subset $S \subset P$ of the points, with $|S| = n/2$, such that S is equiangular, and $SA^+(S, b) = \gamma^{\frac{n}{2}}$.
3. In this last case, we have $W_3 = \langle \gamma, (\alpha, \beta)^{k_1}, \alpha, \gamma, (\beta, \alpha)^{k_2}, \beta \rangle = \langle \beta, (\alpha + \beta), \alpha, (\beta, \alpha)^{k_1}, (\beta + \alpha), (\beta, \alpha)^{k_2} \rangle = \langle 2\gamma, \gamma^{k_1}, \gamma, \gamma^{k_2} \rangle = \langle \delta, \gamma^{\frac{n}{2}-2} \rangle$. That is, there exists a subset $S \subset P$ of the points, with $|S| = n/2 - 1$, such that S is equiangular with one gap, and $SA^+(S, b) = \langle \delta, \gamma^{\frac{n}{2}-2} \rangle$. $\square$

Appendix B. Testing for biangularity. The following routine is used to determine whether a set of points P is biangular with at most two gaps when the two angles *are not given*, and if so find the set B of all centers. Note that, for n robots, this routine needs $O(2^n)$ time to compute the biangularity tests, and hence the computation of the critical points by a robot needs at least $O(2^n)$ time.

Appendix C. Proofs from section 5.3.2.

Proof of Claim 1. We distinguish two cases.

(1.1) One of the two robots in T, say, l'', is not still at time t. By hypothesis, either l'' is acting on l''_m, the median point on $Rad(l'')$, or l'' is performing `movePairwiseCautiously`(l', l'', c).

(a) If l'' is acting on l''_m, then by hypothesis, at time t,

$$|\mathcal{CP}((l', c), (l'', c))| = 1;$$

`Routine3` allows l' to move toward c by `movePairwiseCautiously`(l', l'', c), while all the other robots stay still (on *SEC*). Let t' be the first time when this happens: since by hypothesis the only critical point on the way is c, the destination point of l' can only be c. Moreover, within finite time, say at t'', l'' stops before or at l''_m. Let $t''' > t''$ be the first time l'' becomes active again. If l' is already at c at time t''', then by `Routine4` only l'' is allowed to move; it moves cautiously toward c, and condition (iii) of the claim holds, with l'' acting on c. Otherwise, `Routine3` allows to move only l' and l''; they move toward c by performing `movePairwiseCautiously`(l', l'', c). Note that the movements of l' and l'' can only be along $Rad(l')$ and $Rad(l'')$, respectively; hence, during these movements the configuration remains plain, $SA(\mathcal{R})$ does not change (it stays *simple*), $s = 2$, and the robots in T are the only ones allowed to move. Since there are no critical points on their way to c, by Assumption DIS, within finite time, either both of them reach c simultaneously (condition (ii) holds), or one of them reaches c first, while the other is acting on c (condition(iii)).

Routine 7 Tests for regular biangularity and irregular biangularity.

---

1. **Testing for biangularity (P).**

   Clearly $|P| = n$ must be even. The testing is easily accomplished as follows: for each $S \subset P$, with $|S| = n/2$, determine if both S and $P \setminus S$ are equiangular and their center of equiangularity c is the same; if so, P is biangular and $B = \{c\}$. Since, by Lemma 2.6, the center of biangularity is unique, the testing can stop as soon as the first set is found.

2. **Testing for biangularity with one gap (P).**

   Clearly $|P| = n$ must be odd. For each $S \subset P$, with $|S| = \lceil n/2 \rceil$, determine if it is equiangular; by property 5, such a condition is necessary. If such a S is found, let c be its center of equiangularity; it is now trivial to verify whether P is biangular with a gap with respect to c. If so, P is biangular with a gap, and $c \in B$. All sets so found are clearly biangular with one gap; the necessity expressed by property 5 ensures that all such sets are found, and thus B contains all the centers of biangularity with one gap of P.

3. **Testing for biangularity with two gaps (P).**

   Clearly $|P| = n$ must be even. The testing can be accomplished in two successive phases as follows. In the first phase, for each $S \subset P$, with $|S| = n/2$, determine if it is equiangular; if so, let c be its center of equiangularity; it is now trivial to verify whether P is biangular with two gaps with respect to c, and if so, $c \in B$. In the second phase, for each $S \subset P$, with $|S| = n/2 - 1$, determine if both S and $P \setminus S$ are equiangular with one gap and if they share the same center of equiangularity c; if so, $c \in B$. All sets so found are clearly biangular with two gaps; the necessity expressed by property 6 ensures that all such sets are found, and thus B contains all the centers of biangularity with two gaps of P.

4. **Testing for irregular biangularity (P).**

   Clearly $|P| = n$ must be even. For each $p \in P$ test whether p is the center of biangularity with one gap for $P \setminus \{p\}$. If so, P is irregular biangular and $B = \{p\}$. Since, by Lemma 2.6, the center of irregular biangularity is unique, the testing can stop as soon as the first such point is found.

(b) If l'' is performing `movePairwiseCautiously`(l', l'', c), then by `Routine3` only l' and l'' are allowed to move by`movePairwiseCautiously`(l', l'', c). Let $t' > t$ be the first time l'' stops. Since the movements of l' and of l'' can be only along $Rad(l')$ and $Rad(l'')$, respectively, during these movements the configuration remains plain and $SA(\mathcal{R})$ does not change; hence, it is still *simple*, $s = 2$. By Properties 10 and 11, at time t' one of the conditions of the claim holds.

(1.2) Both robots in T are still. By `Routine3`, only l' and l'' are allowed to move, by performing `movePairwiseCautiously`(l', l'', c). Without loss of generality, let l' be the first robot to stop after a non null movement at a time $t' > t$. The proof follows as in previous Case (1.1).b.

Proof of Claim 2. Let $N'(t)$ and $N''(t)$ be the number of critical points ahead

of l' and l'', respectively, at time t. Recall that, by Property 10, when executing `movePairwiseCautiously`(l', l'', c) at time t, l' (resp., l'') will perform a nonnull movement only if $|N'(t)| \geq |N''(t)|$ (resp., $|N''(t)| \geq |N'(t)|$). We distinguish three cases, depending on which robot, if any, is acting on c.

(2.1) Let l'' be not still at time t. By hypothesis, l'' is either acting on l''_m, the median point on $Rad(l'')$, or is performing `movePairwiseCautiously`(l', l'', c). In the first case, by hypothesis, at time t, $|\mathcal{CP}((l', c), (l'', c))| = 1$. According to `Routine3`, l' does not move until l'' enters SEC, say at time $t' > t$, by executing `moveTo`(l''_m). As soon as l'' enters SEC, the Claim hold. In the second case, l'' is heading toward c by performing `movePairwiseCautiously`(l', l'', c). First note that in this case, at time t, $|\mathcal{CP}((l', c), (l'', c))| > 1$: in fact, by `Routine3`, starting from the rim of the SEC l'' can never execute `movePairwiseCautiously`(l', l'', c) if $|\mathcal{CP}((l', c), (l'', c))| = 1$. Also, since by hypothesis l'' is not still, by Property 10, $N''(t) \geq N'(t)$; and, by definition of `movePairwiseCautiously`(l', l'', c), l' will not move as long as $N''(t) > N'(t)$. Two cases arise, depending on the values of $N'(t)$ and $N''(t)$.

(a) $N''(t) = N'(t) > 1$ at time t. In this case, the algorithm forces both l' and l'' to perform `movePairwiseCautiously`(l', l'', c). By Properties 10 and 11, the two robots are forced to move toward c in lock-step, and within finite time both robots will be inside SEC and still on a pair of critical points, and the claim holds. Note that at this time, the configuration can be either regular biangular or not.

(b) $N''(t) > N'(t) > 1$ at time t. In this case, by Properties 10 and 11, l'' is the only robot allowed to move, by `movePairwiseCautiously`(l', l'', c). The claim holds as soon as l'' enters SEC.

(2.2) Let l' be not still at time t; by hypothesis, l' is (a) either acting on l'_m, the median point on $Rad(l')$, while l'', on SEC, is still and $|\mathcal{CP}((l', c), (l'', c))| = 1$; or (b) it is heading toward c by performing `movePairwiseCautiously`(l', l'', c), with l'' on SEC and still.

In case (a), note that, if l' stops while l'' is on SEC, by `Routine3`, l' will wait until l'' enters SEC. Let $t' \geq t$ be the first time when l'' becomes active: l'' is obliged to perform `moveTo`(l''_m), with l''_m the median point of $Rad(l'')$. Now, let $t'' > t'$ be the first time when l'' enters SEC. At this time, either l' is still, or it is acting on l'_m. In the first case, the claim follows. In the second case, the first time either l' or l'' stop, the claim follows as well.

In case (b), l' is performing `movePairwiseCautiously`(l', l'', c). According to `Routine3`, at time t, $|\mathcal{CP}((l', c), (l'', c))| > 1$: in fact, if $|\mathcal{CP}((l', c), (l'', c))| = 1$, l' can never execute `movePairwiseCautiously`(l', l'', c). Also, since by hypothesis l' is not still, by Property 10, $N'(t) \geq N''(t)$; and, by definition of `movePairwiseCautiously`(l', l'', c), l'' will not move as long as $N'(t) > N''(t)$. The proof follows similarly to previous cases (2.1).a and (2.1).b.

(2.3) Both robots are still at time t. Three cases arise, depending on the value of $|\mathcal{CP}((l', c), (l'', c))|$.

(a) If $|\mathcal{CP}((l', c), (l'', c))| = 1$, then the only critical point for both l' and l'' is c. In this case, `Routine3` does not allow l' to move until l'' enters SEC. Moreover, l'' can only perform `moveTo`(l''_m). Therefore, the first time when l'' starts executing the move operation, the claim holds.

(b) If $|\mathcal{CP}((l', c), (l'', c))| > 1$, let us assume without loss of generality that

$N'(t) \geq N''(t)$. If $N'(t) = N''(t) > 1$ then the proof follows as in previous case (2.1).a. If $N'(t) > N''(t) > 1$, then the proof follows similarly to previous case (2.1).b. Finally, if $N'(t) > N''(t) = 1$, then $|\mathcal{CP}((l', c), (l'', c))| > 1$. In this case, the algorithm forces both l' and l'' to perform `movePairwise-Cautiously`(l', l'', c). By Properties 10 and 11, l' is the only one allowed to move (toward c) as long as no regular biangular configuration is formed (l'' could be on a critical point on SEC) and $N' > 1$. By Assumption DIS, within finite time, say, at t', $N'(t') = N''(t') = 1$. At this time `Routine3` forces l' to not move (at time t', l'' is on the SEC), and forces l'' to move toward the median point on $Rad(l'')$. The first time l'' enters SEC, the claim holds.

Note that, in all the above arguments, the movements of both robots can be only along $Rad(l')$ and $Rad(l'')$, respectively; hence, during these movements the configuration remains plain, $SA(\mathcal{R})$ does not change (it stays *simple*), $s = 2$, and the robots in T are the only ones allowed to move.

Appendix D. Proofs From section 5.3.3.

Proof of Lemma 5.11. If all robots are on SEC, a unique class T of $\frac{n}{k}$ robots is elected by `ElectClass()`; by Lemma 2.4, SEC remains invariant if the robots of this class are removed.

By `Routine3`, the only robots that can move are those in T: the first move a robot from the class performs while they are all on SEC, is on the midpoint of its radius; hence, in this move no robot can reach c. Let t be the first time a robot from T leaves SEC; if more than one robots does so, the lemma follows. If only one robot does so, say, r, the other robots from T can move either toward c (they observed when r was on the midpoint of its radius, or on the way toward it) or the median point of the radius where they lie (they observed when r was on SEC). Note that, if there is only one robot from the class inside SEC, by `Routine3`, it does not move until at least another robot from T enters SEC. As soon as the first of them leaves SEC, by Lemma 2.5 and since the movements are only on radii of SEC, s does not change, and the lemma holds.   ☐

Proof of Lemma 5.12. Notice that T coincides with the class selected by `ElectClass()` at time t. Let us denote by $\mathcal{A}$ the set of robots that are acting at time t. By hypothesis, at time t each robot in $\mathcal{A}$ is acting on either c or the median point of its radius; let us denote by $\mathcal{A}_m$ the subset of $\mathcal{A}$ that contains the robots that are acting on the median point of their radius. First note that, by definition of `Routine3`, the robots that are acting on c move independently from the robots that are acting on the median points of their radii; in particular, only the robots that are acting on c can reach it; furthermore, by Lemma 2.5 and since the movements are only on radiii of SEC, no regular biangular configuration can be formed while those robots are moving toward c, and a dense point can only be created at c.

Let us consider now the time $t' > t$ when the first of the robots in $\mathcal{A}_m$ stops (before or at the median point on its radius); at this time, the distance of r' from c has decreased. We distinguish three cases:

1. If at t' no robot is at c, since all robots move radially between time t and t', $SA(\mathcal{R})$ stays simple with $s > 2$, and the period of $SA(\mathcal{R})$ does not change. When r' starts acting again, by `Routine3` it will be safely acting on c.
2. If at t' c is dense, then by Algorithm GOGATHER, r' starts safely acting on c.
3. If at t' one robot is at c, since all robots move radially between time t and t', then $SA(\mathcal{R} \setminus \{c\})$ is simple, and $\mathcal{R}$ is irregular periodic. By `Routine4`, r'

starts safely acting on c.

Hence, in all three cases, $|\mathcal{A}_m|$ decreases by one unit. Since, by `Routine3` and `Routine4`, all movements are only on radii of *SEC*, and, by Lemma 2.5 no regular biangular configuration can be created during any movement of the robots, cases 1, 2, and 3 above hold for every robot in $\mathcal{A}_m$. Hence, the lemma follows the first time the last robot in $\mathcal{A}_m$ stops. $\square$

Proof of Lemma 5.13. We proceed by induction on the number $m \geq 1$ of classes with at least one robot inside *SEC*. The lemma trivially holds for $m = 1$. Let it hold for $1 \leq m \leq k$ classes, and consider the case when inside *SEC* there are robots from $k + 1$ classes. According to `Routine3`, only robots from the same class T are allowed to move—the class chosen by `SelectClassInside()`. We now prove that in finite time the number of classes inside *SEC* becomes k; this is done by induction on the number T_{IN} of robots in T that are inside *SEC*.

Let $T_{IN} = 1$; let $r \in T$ be the only robot of T inside *SEC*. By `Routine3`, r is the only robot allowed to move by executing `moveTo(`*SEC*`)`, and if it is already moving, by hypothesis it is moving toward *SEC*. Let $t' > t$ be the first time when r stops after a nonnull movement. Since the movement of r is only along the radius where it lies, during r's movement the configuration remains plain and $SA(\mathcal{R})$ does not change; hence, it is still *simple* and $s > 2$. When r stops, either (i) r is on *SEC*, or (ii) r is at a point closer to *SEC*. In case (i), at time t' there are k classes inside *SEC*. In case (ii), the hypotheses of the lemma are still met and the distance of r from *SEC* has decreased. Hence, by Assumption DIS, case (i) will eventually hold and there will be only k classes inside *SEC*.

Let us assume the lemma holds for $T_{IN} = h$, and consider the case when the number of robots from T that are inside *SEC* is $h + 1$. Let $T^* \subseteq T$ be the set of robots that first stop after executing `moveTo(`*SEC*`)` with a nonnull movement, and let $t' > t$ be the first time when this happens; note that $|T^*| \geq 1$. Since the movements of robots in T^* are only along the radii where they lie, during their movement the configuration remains plain and $SA(\mathcal{R})$ does not change; hence, it is still *simple* and $s > 2$. When robots in T^* stops, either (iii) at least one of them reached *SEC*, or (iv) none of them reached *SEC*. In case (iii), at time t' there are at most h robots from T^* inside *SEC*; hence, by induction, in finite time there are k classes inside *SEC*. In case (iv), since the hypotheses of the lemma are still met and the distance of the robots in T^* from *SEC* has decreased, by Assumption DIS, case (iii) will eventually hold and there will be only k classes inside *SEC*.

Hence the lemma follows. $\square$

REFERENCES

[1] N. AGMON AND D. PELEG, *Fault-tolerant gathering algorithms for autonomous mobile robots*, SIAM J. Comput., 36 (2006), pp. 56–82.

[2] L. ANDEREGG, M. CIELIEBAK, AND G. PRENCIPE, *Efficient algorithms for detecting regular point configurations*, in Proceedings of the 9th Italian Conference on Theoretical Computer Science (ICTCS), Lecture Notes in Comput. Sci. 3701, Springer, Berlin, 2005, pp. 23–35.

[3] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, *A distributed memoryless point convergence algorithm for mobile robots with limited visibility*, IEEE Trans. Robotics Automat., 15 (1999), pp. 818–828.

[4] C. Bajaj, *The algebraic degree of geometric optimization problems*, Discrete Comput. Geom., 3 (1988), pp. 177–191.

[5] F. Bullo, J. Cortes, and S. Martinez, *Distributed algorithms for robotic networks*, in Encyclopedia of Complexity and Systems Science, R. Meyers, ed., Springer-Verlag, New York, 2009, pp. 7712–7728.

[6] I. Chatzigiannakis, M. Markou, and S. Nikoletseas, *Distributed circle formation for anonymous oblivious robots*, in Proceedings of the 3rd International Workshop on Experimental and Efficient Algorithms (WEA), Lecture Notes in Comput. Sci. 3059, Springer, Berlin, 2004, pp. 159–174.

[7] S. G. Chaudhuri and K. Mukhopadhyaya, *Gathering asynchronous transparent fat robots*, in Proceedings of the 6th International Conference on Distributed Computing and Internet Technology (ICDCIT), Lecture Notes in Comput. Sci. 5966, Springer, Berlin, 2010, pp. 170–175.

[8] M. Cieliebak, *Gathering non-oblivious mobile robots*, in Proceedings of the 6th Latin American Conference on Theoretical Informatics (LATIN), Lecture Notes in Comput. Sci. 2976, Springer, Berlin, 2004, pp. 577–588.

[9] M. Cieliebak and G. Prencipe, *Gathering autonomous mobile robots*, in Proceedings of the 9th International Colloquium On Structural Information And Communication Complexity (SIROCCO), Mont Saint-Michel, France, 2002, pp. 57–72.

[10] E. J. Cockayne and Z. A. Melzak, *Euclidean constructibility in graph-minimization problems*, Math. Mag., 42 (1969), pp. 206–208.

[11] R. Cohen and D. Peleg, *Convergence properties of the gravitational algorithms in asynchronous robot systems*, SIAM J. Comput., 34 (2005), pp. 1516–1528.

[12] R. Cohen and D. Peleg, *Convergence of autonomous mobile robots with inaccurate sensors and movements*, in Proceedings of the 23rd Symposium on Theoretical Aspects of Computer Science (STACS), Springer, Berlin, 2006, pp. 549–560.

[13] R. Cohen and D. Peleg, *Local spreading algorithms for autonomous robot systems*, Theoret. Comput. Sci., 399 (2008), pp. 71–82.

[14] A. Cord-Landwehr, B. Degener, M. Fischer, M. Hüllmann, B. Kempkes, A. Klaas, P. Kling, S. Kurras, M. Mrtens, F. Meyer auf der Heide, C. Raupach, K. Swierkot, D. Warner, C. Weddemann, and D. Wonisch, *Collision-less gathering of robots with an extent*, in Proceedings of the 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM), Lecture Notes in Comput. Sci. 6543, Springer, Berlin, 2011, pp. 178–189.

[15] A. Cord-Landwehr, B. Degener, M. Fischer, M. Hüllmann, B. Kempkes, A. Klaas, P. Kling, S. Kurras, M. Mrtens, F. Meyer auf der Heide, C. Raupach, K. Swierkot, D. Warner, C. Weddemann, and D. Wonisch, *A new approach for analyzing convergence algorithms for mobile robots*, in Proceedings of the 38th International Colloquium on Automata, Languages and Programming (ICALP), Lecture Notes in Comput. Sci. 6756, Springer, Berlin, 2011, pp. 650–661.

[16] J. Czyzowicz, L. Gasieniec, and A. Pelc, *Gathering few fat mobile robots in the plane*, Theoret. Comput. Sci., 410 (2009), pp. 481–499.

[17] S. Das, P. Flocchini, N. Santoro, and M. Yamashita, *On the computational power of oblivious robots: Forming a series of geometric patterns*, in Proceedings of the 29th Annual ACM Symposium on Principles of Distributed Computing (PODC), ACM, New York, 2010, pp. 267–276.

[18] X. Défago and S. Souissi, *Non-uniform circle formation algorithm for oblivious mobile robots with convergence toward uniformity*, Theoret. Comput. Sci., 396 (2008), pp. 97–112.

[19] B. Degener, B. Kempkes, T. Langner, F. Meyer auf der Heide, P. Pietrzyk, and R. Wattenhofer, *A tight runtime bound for synchronous gathering of autonomous robots with limited visibility*, in Proceedings of the 23rd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), ACM, New York, 2011, pp. 139–148.

[20] S. Devismes, F. Petit, and S. Tixeuil, *Optimal probabilistic ring exploration by semisynchronous oblivious robots*, in Proceedings of the 16th Colloquium on Structural Information and Communication Complexity (SIROCCO), Lecture Notes in Comput. Sci. 5869, Springer-Verlag, Berlin, 2009, pp. 195–208.

[21] Y. Dieudonné, S. Dolev, F. Petit, and M. Sega, *Deaf, dumb, and chatting asynchronous robots*, in Proceedings of the 13th International Conference on Principles of Distributed Systems (OPODIS), Springer, Berlin, 2009, pp. 71–85.

[22] Y. Dieudonné, O. Labbani-Igbida, and F. Petit, *Circle formation of weak mobile robots*, ACM Trans. Autonomous and Adaptive Systems, 3 (2008), pp. 1–20.

[23] S. Durocher and D. Kirkpatrick, *The projection median of a set of points*, Comput. Geom., 42 (2009), pp. 364–375.

[24] A. Efrima and D. Peleg, *Distributed models and algorithms for mobile robot systems*, in Proceedings of the 33rd International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM), Lecture Notes in Comput. Sci. 4362, Springer-Verlag, Berlin, 2007, pp. 70–87.

[25] P. Flocchini, G. Prencipe, and N. Santoro, *Computing by Mobile Robotic Sensors*, in Theoretical Aspects of Distributed Computing in Sensor Networks, S. Nikoletseas and J. Rolim, eds., Monogr. Theoret. Comput. Sci., Springer-Verlag, Berlin, 2011, pp. 655–693.

[26] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer, *Gathering of robots with limited visibility*, Theoret. Comput. Sci., 337 (2005), pp. 147–168.

[27] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer, *Arbitrary pattern formation by asynchronous oblivious robots*, Theoret. Comput. Sci., 407 (2008), pp. 412–447.

[28] J. Fredslund and M. Matarić, *A general algorithm for robot formations using local sensing and minimal communication*, IEEE Trans. Robotics Automation, 18 (2002), pp. 837–846.

[29] A. Ganguli, J. Cortés, and F. Bullo, *Multirobot rendezvous with visibility sensors in nonconvex environments*, IEEE Trans. Robotics, 25 (2009), pp. 340–352.

[30] N. Gordon, Y. Elor, and A. M. Bruckstein, *Gathering multiple robotic agents with crude distance sensing capabilities*, in Proceedings of the 6th International Conference on Ant Colony Optimization and Swarm Intelligence, Lecture Notes in Comput. Sci. 5217, Springer, Berlin, 2008, pp. 72–83.

[31] N. Heo and P. K. Varshney, *Energy-efficient deployment of intelligent mobile sensor networks*, IEEE Trans. Systems Man CyberNetics A Systems Humans, 35 (2005), pp. 78–92.

[32] A. Howard, M. J. Mataric, and G. S. Sukhatme, *An incremental self-deployment algorithm for mobile sensor networks*, Autonomous Robots, 13 (2002), pp. 113–126.

[33] Y. Ikemoto, Y. Hasegawa, T. Fukuda, and K. Matsuda, *Gradual spatial pattern formation of homogeneous robot group*, Inform. Sci., 171 (2005), pp. 431–445.

[34] T. Izumi, S. Souissi, Y. Katayama, N. Inuzuka, X. Défago, K. Wada, and M. Yamashita, *The gathering problem for two oblivious mobile robots with unreliable compasses*, SIAM J. Comput., 41 (2012), pp. 26–46.

[35] A. Kansal, W. Kaiser, G. Pottie, M. Srivastava, and G. S. Sukhatme, *Reconfiguration methods for mobile sensor networks*, ACM Trans. Sensor Networks, 3 (2007), pp. 22–23.

[36] Y. Katayama, Y. Tomida, H. Imazu, N. Inuzuka, and K. Wada, *Dynamic compass models and gathering algorithms for autonomous mobile robots*, in Proceedings of the 14th Colloquium on Structural Information and Communication Complexity (SIROCCO), Lecture Notes in Comput. Sci. 4474, Springer, Berlin, 2007, pp. 274–288.

[37] R. Klasing, E. Markou, and A. Pelc, *Gathering asynchronous oblivious mobile robots in a ring*, Theoret. Comput. Sci., 390 (2008), pp. 27–39.

[38] D. R. Kowalski and A. Malinowski, *How to meet in anonymous network*, Theoret. Comput. Sci., 399 (2008), pp. 141–156.

[39] Y. Kupitz and H. Martini, *Geometric aspects of the generalized Fermat-Torricelli problem*, Intuitive Geometry, 6 (1997), pp. 55–127.

[40] J. Lee, S. Venkatesh, and M. Kumar, *Formation of a geometric pattern with a mobile wireless sensor network*, J. Robotic Systems, 21 (2004), pp. 517–530.

[41] J. Lin, A. S. Morse, and B. D. O. Anderson, *The multi-agent rendezvous problem. Part 1: The synchronous case*, SIAM J. Control Optim., 46 (2007), pp. 2096–2119.

[42] J. Lin, A. S. Morse, and B. D. O. Anderson, *The multi-agent rendezvous problem. Part 2: The asynchronous case*, SIAM J. Control Optim., 46 (2007), pp. 2120–2147.

[43] Z. Lin, B. Francis, and M. Maggiore, *Getting mobile autonomous robots to rendezvous*, in Workshop on Control of Uncertain Systems, Lecture Notes in Control and Information Sciences 329, Springer-Verlag, Berlin, 2006, pp. 119–137.

[44] S. Martínez, *Practical multiagent rendezvous through modified circumcenter algorithms*, Automatica J. IFAC, 45 (2009), pp. 2010–2017.

[45] S. Martínez, F. Bullo, J. Cortes, and E. Frazzoli, *On synchronous robotic networks—parts I and II*, IEEE Transactions on Automatic Control, 52 (2007), pp. 2199–2226.

[46] Y. Oasa, I. Suzuki, and M. Yamashita, *A robust distributed convergence algorithm for autonomous mobile robots*, in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, IEEE, Washington, DC, 1997, pp. 287–292.

[47] L. E. PARKER, *Current state of the art in distributed autonomous mobile robotics*, in Distributed Autonomous Robotic Systems 4, L. E. Parker, G. Bekey, and J. Barhen, eds., Springer, Berlin, 2000, pp. 3–12.

[48] G. PRENCIPE, *Impossibility of gathering by a set of autonomous mobile robots*, Theoret. Comput. Sci., 384 (2007), pp. 222–231.

[49] S. SOUISSI, X. DÉFAGO, AND M. YAMASHITA, *Using eventually consistent compasses to gather memory-less mobile robots with limited visibility*, ACM Trans. Autonomous Adaptive Systems, 4 (2009), pp. 1–27.

[50] O. SOYSAL, E. BAHÇECI, AND E. ŞAHIN, *Aggregation in swarm robotic systems: Evolution and probabilistic control*, Turkish J. Electrical Engineering, 15 (2007), pp. 199–225.

[51] K. SUGIHARA AND I. SUZUKI, *Distributed algorithms for formation of geometric patterns with many mobile robots*, J. Robotics Systems, 13 (1996), pp. 127–139.

[52] I. SUZUKI AND M. YAMASHITA, *Distributed anonymous mobile robots: Formation of geometric patterns*, SIAM J. Comput., 28 (1999), pp. 1347–1363.

[53] E. WEISZFELD, *Sur le point pour lequel la somme des distances de n points donnés est minimum*, Tohoku Math., 43 (1936), pp. 355–386.

[54] E. WELZL, *Smallest enclosing disks (balls and ellipsoids)*, in New Results and New Trends in Computer Science, Springer, Berlin, 1991, pp. 359–370.

[55] DÉFAGO X, M. GRADINARIU, S. MESSIKA, AND P. RAIPIN-PARVÉDY, *Fault-tolerant and self-stabilizing mobile robots gathering*, in Proceedings of the 20th International Symposium on Distributed Computing (DISC), Lecture Notes in Comput. Sci. 4167, Springer, Berlin, 2006, pp. 46–60.

[56] M. YAMASHITA AND I. SUZUKI, *Characterizing geometric patterns formable by oblivious anonymous mobile robots*, Theoret. Comput. Sci., 411 (2010), pp. 2433–2453.

[57] J. YU, M. LAVALLE, AND D. LIBERZON, *Rendezvous without coordinates*, in Proceedings of the 47th IEEE Conference on Decision and Control, IEEE Control Systems Society, Piscataway, NJ, 2008, pp. 1803–1808.